

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky

DIPLOMOVÁ PRÁCE

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra kybernetiky a biomedicínského inženýrství

Vzdálená aktualizace řídicích systémů s mikrokontroléry
Remote Update of Control Systems with Microcontrollers

Zadání diplomové práce

Student:

Bc. Michal Pross

Studijní program:

N2649 Elektrotechnika

Studijní obor:

2601T004 Měřicí a řídicí technika

Téma:

Vzdálená aktualizace řídicích systémů s mikrokontroléry
Remote Update of Control Systems with Microcontrollers

Zásady pro vypracování:

Diplomová práce se zabývá návrhem a realizací jednotky, u které lze vzdáleně po síti (ethernet) modifikovat firmware uložený v paměti mikrokontroléru. Jednotka se skládá ze základních komponent používaných pro měření a regulace.

V souhrnu je práce charakterizována následujícími body:

1. Rešerše možností změny firmware v mikrokontrolérech.
2. Návrh a realizace řešení pro vzdálenou aktualizaci firmware pro mikrokontroléry.
3. Návrh a realizace vývojové desky pro ověření použité metody.
4. Test řešení v rámci jednoduché regulační úlohy.
5. Návrh a implementace software a knihoven pro cílová zařízení pro konfiguraci vývojové desky.
6. Zhodnocení dosažených výsledků.

Seznam doporučené odborné literatury:

- [1] BRTNÍK, Bohumil a David MATOUŠEK. *Mikroprocesorová technika: [práce s mikrokontroléry řady ATMEGA AVR ATXmega A4]*. 1. vyd. Praha: BEN - technická literatura, 2011. ii, [150] s. ISBN 978-80-7300-406-4.
- [2] PUŽMANOVÁ, Rita. *TCP/IP v kostce*. 2. upr. a rozš. vyd. České Budějovice: Kopp, 2009. 619 s. ISBN 978-80-7232-388-3.
- [3] CADY, Frederick M. *Microcontrollers and Microcomputers Principles of Software and Hardware Engineering*. 2nd ed. New York: Oxford University Press, 2010. xii, 477 p. ISBN 01-953-7161-5.
- [4] MAZIDI, Muhammad A., Sarmad NAIMI a Sepehr NAIMI. *AVR Microcontroller and Embedded Systems: Using Assembly and C*. Upper Saddle River, N.J.: Prentice Hall, c2011. xiv, 776 p. ISBN 01-380-0331-9.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **Ing. Zdeněk Slanina, Ph.D.**

Datum zadání: 16.11.2012

Datum odevzdání: 07.05.2013



doc. Ing. Jiří Koziorek, Ph.D.
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.
děkan fakulty

PROHLÁŠENÍ

„Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.“

Datum 7.5.2013

Podpis



PODĚKOVÁNÍ

Děkuji vedoucímu mé diplomové práce Ing. Zdeňku Slaninovi, Ph.D. a Bc. Aleši Kurečkovi za pomoc a podporu při řešení diplomové práce.

ABSTRAKT

Diplomová práce se zabývá architekturou mikrokontrolérů AVR a možnostmi, jak lze tuto rodinu mikrokontrolérů naprogramovat. Pojednává o návrhu a realizaci vývojové desky, která simuluje vestavěný řídicí systém, na kterém se bude provádět vzdálená aktualizace firmware.

Nejprve jsou vytvořeny knihovny do programu Eagle, pomocí kterého je nakresleno schéma a DPS vývojové desky. Kritéria pro návrh jsou dána z předchozí analýzy řešení. Je vytvořen také software pro jednoduché nastavení periférií a MCU. Do vývojového prostředí AVR Studio se importuje nastavení periférií a vytvoří se firmware. Samotná aktualizace probíhá pomocí programů AVR Studio, HW VSP3 a hardware modulu NanoSocketLAN a JTAG programátoru.

ABSTRACT

This thesis deals with the architecture of AVR microcontrollers and the possibilities of how this family of microcontrollers program. Discusses the design and implementation of development board, which simulates the embeded control system which will perform remote firmware updates.

At first libraries are created in Eagle program which is used to drae schematic and PCB development board. Criteria for the design are given in the previous solution analysis. Also a software to simply set peripherals and MCU is created. Peripherals settings is imported into AVR Studio development environment and firmaware is created. The update itself is using AVR Studio, HW VSP3 and hardware module NanoSocketLAN and JTAG programmer.

KLÍČOVÁ SLOVA

AVR, ATmega, Bootloader, DPS, Ethernet, Firmware, JTAG, Mikrokontrolér, NanoSocketLAN, Vzdálená aktualizace

KEY WORDS

AVR, ATmega, Bootloader, DPS, Ethernet, Firmware, JTAG, Microcontroller, NanoSocketLAN, Remote update

Seznam použitých symbolů a zkratk

3DES	3 Data Encryption Standard
AES	Advanced Encryption Standard
ARP	Address Resolution Protocol
ASIC	Application-specific integrated circuit
AVR	Alf Vegard Risc procesor
BPS	Bit Per Second
CD	Copmact Disc
COM	Computer Output on Microfilm
CPLD	Complex Programmable Logic Device
DHCP	Dynamic Host Configuration Protocol
DIL	Dual In Line
DPS	Deska plošného spoje
EEPROM	Electrically Erasable Programmable Read-Only Memory
FPGA	Field Programmable Gate Array
FPSLIC	Field Programmable System Level Integrated Circuits
FTDI	Future Technology Devices International
FTP	File Transfer Protocol
FTPS	File Transfer Protocol Secure
FW	Firmware
GCC	GNU C Compiler
GND	Ground
GNU	Gnu's Not Unix
HAL	Hot Air Levelling
HEX	Hexadecimal
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure
HVP	High-voltage programming
HW	Hardware
Hz	Kmitočet [hertz]
I/O	Input/ Output
ICMP	Internet Control Message Protocol
IP	Internet Protocol
ISP	In-system programming
JTAG	Joint Test Action Group
LAN	Local Area Network
LED	Light Emitting Diode (Světloemitující dioda)
LPM	Load Program Memory
MCU	Micro Controller Unit

MISO	Master In – Slave Out
MOSI	Master Out – Slave In
PC	Personal Computer (osobní počítač)
Pin	Vývod integrovaného obvodu
Px	Pixel
RGB	Red Green Blue
RISC	Reduced Instruction Set Computing (redukovaná instrukční sada procesoru)
RXD	Receive Data
SCK	Serial Clock
SHA	Secure Hash Algorithm
SPI	Serial Peripheral Interface Bus
SPM	Store Program Memory
TCK	Test Clock Input
TCP/IP	Transmission Control Protocol/Internet Protocol
TDI	Test Data Input
TDO	Test Data Output
TMS	Test Mode Select Input
TQFP	Thin Quad Flat Package
TRST	Test Reset Input
TWI	Two Wire Interface
TXD	Transmit Data
UART	Universal Asynchronous Receiver Transmitter
USB	Universal Serial Bus
V	Elektrické napětí [volt]
VŘS	Vestavěný řídicí systém
XAML	Extensible Application Markup Language

Obsah

1	Úvod	1
2	Architektura Mikrokontrolérů AVR	2
3	Možnosti změny firmware v mikrokontrolérech	4
3.1	Sériové programování	4
3.2	Paralelní programování	5
3.3	JTAG (Joint Test Action Group)	5
3.4	Bootloader	6
3.5	DebugWIRE	7
4	Návrh řešení pro vzdálenou aktualizaci firmware	8
4.1	Ethernetový modul NanoSocketLAN	9
4.1.1	Podporované protokoly	10
4.1.2	Přenosové rychlosti	10
4.1.3	Schéma zapojení vývodů	10
4.1.4	UART rozhraní	11
4.2	JTAG programátor	12
4.2.1	Schéma JTAG programátoru	12
4.2.2	DPS JTAG programátoru	13
5	Vzdálená aktualizace firmware pomocí PC	15
5.1	Instalace software	15
5.2	Nastavení SW a HW	15
5.2.1	Nastavení FT232	15
5.2.2	Nastavení NanoSocketLAN	16
5.2.3	Nastavení HW VSP3 (Virtual Serial Port)	21
5.2.4	Aktualizace firmware JTAG programátoru	23
5.2.5	Nastavení AVR Studia	24
5.2.6	Typická ukázka práce s vytvořeným hardware	24
6	Návrh vývojové desky	26
6.1	Požadavky na vybavení vývojové desky	26

6.2	Požadavky na výrobu vývojové desky	27
6.3	Schéma vývojové desky	27
6.3.1	Zdroj	27
6.3.2	Přepínač signálu	28
6.3.3	Mikrokontroléry ATmega	29
6.3.4	LCD displej	29
6.3.5	Bezzákmitové přepínače a tlačítka	29
6.3.6	RS 485	30
6.3.7	Logická sonda	31
6.3.8	USB-UART	32
7	Test řešení v rámci regulační úlohy	33
8	Návrh knihoven a software pro vývojovou desku	34
8.1	Software pro generování obslužných rutin a nastavení vývojové desky	34
8.2	Web server s nápovědou	36
9	Závěr	37
10	Použitá literatura	38
11	Seznam příloh	40

1 Úvod

Tato diplomová práce se zabývá architekturou mikrokontrolérů AVR a možnostmi, jak lze tuto rodinu mikrokontrolérů naprogramovat.

Jsou zde uvedeny základní možnosti programování a lazení MCU. Po srovnání jednotlivých variant je vybráno programování pomocí rozhraní JTAG.

Rozhraní JTAG umožňuje nejen naprogramování MCU, ale také lazení firmware, vyčítání a zapisování do registrů.

Vzdálená aktualizace je řešena naprogramováním mikrokontroléru pomocí JTAG programátoru přímo z vývojového prostředí AVR studio. Propojení mezi AVR Studiem a JTAG programátorem zajišťuje program HW VSP3 a modul NanoSocketLAN, který vytvoří virtuální TCP/IP COM port umožňující přenos sériových dat přes virtuální sériový port, který se chová jako běžný hardwarový COM port.

Dále je vytvořena vývojová deska obsahující všechny potřebné prvky pro vzdálenou aktualizaci firmware MCU, ale také další periferie pro simulaci celého VŘS.

Vývojová deska je nakreslena v programu Eagle pomocí nově vytvořených přehledných knihoven. Je do něj také přidáno menu pro jednodušší návrh DPS.

K jednoduché obsluze vývojové desky a jejího programování byl vytvořen software Mipi Studio, ve kterém lze velice snadno nastavit základní parametry MCU a také generovat obslužné rutiny pro periferie na ní obsažené.

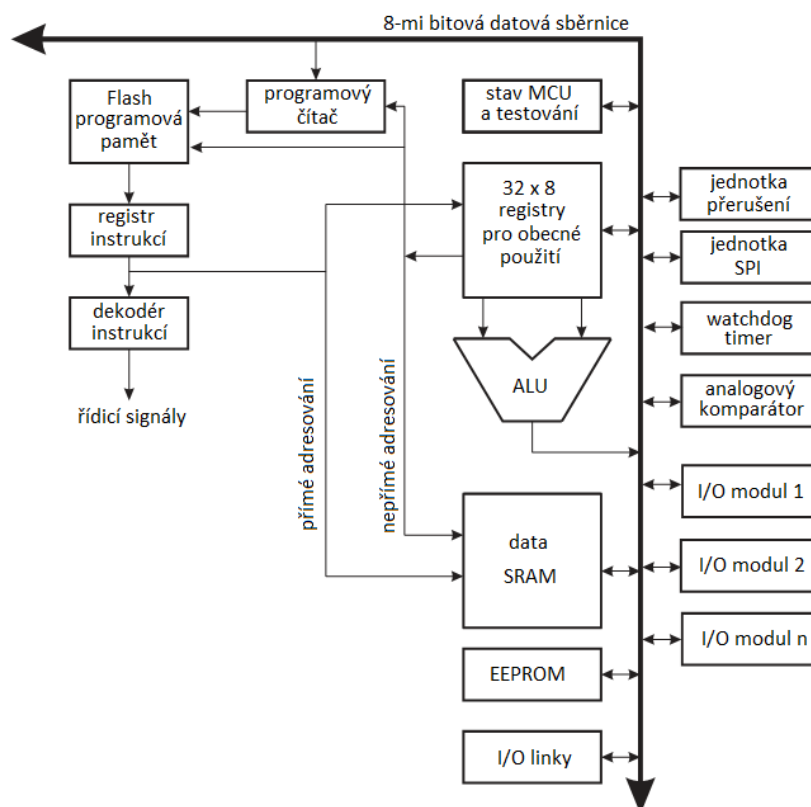
2 Architektura Mikrokontrolérů AVR

AVR je označení pro rodinu 8 - bitových a některých 32 - bitových mikrokontrolérů typu RISC s harvardskou architekturou od firmy Atmel. Za zrodem architektury AVR stojí studenti Alf-Egil Bogen a Vegard Wollan z Norského technologického institutu. Na trhu se tyto mikroprocesory začaly objevovat od roku 1997.

Původní mikropočítač AVR byl vyvinut v Trondheimu v centru ASIC (Aplikačně Specifické Integrované Obvody) zmíněnými studenty, kteří tam pracovali. Dnes je tato firma známá jako Nordic Semiconductor. Při designu úzce spolupracovali na tvorbě kompilátoru, který tak měl mít co nejefektivnější sadu instrukcí.

Mezi základní typy patří řada AT90, která se dnes již nevyrábí. ATtiny, ATmega, ATxmega, FPSLIC (AVR s FPGA) a 32 bitové AVR. [1]

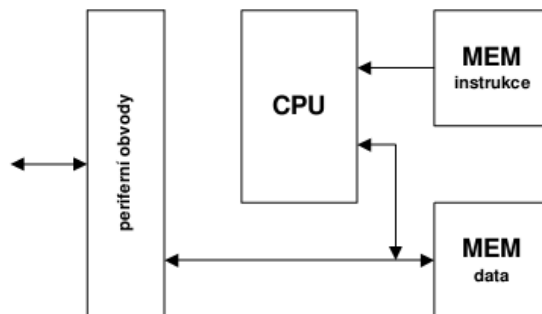
Typickou architekturu mikropočítače rodiny AVR ukazuje obr. č. 1. Jedná se o harvardskou architekturu využívající redukovanou instrukční sadu RISC.



Obr. 1 - Typická architektura AVR[5]

Harvardská architektura je počítačová architektura, která fyzicky odděluje paměť programu a dat a jejich spojovací obvody. Rozdělení datové sběrnice (obrázek č.2) umožní v době zpracovávání jedné instrukce čtení instrukce jiné. Tím lze dosáhnout vyšší rychlosti běhu programu a tedy vyššího

výpočetního výkonu. Lze tedy snadno u této architektury dosáhnout překrývaného provádění instrukcí tzv. pipeline. [2]



Obr. 2 - Sběrnice k přenosu instrukcí a dat v harvard architektuře

RISC označuje procesory s redukovanou instrukční sadou, jejichž návrh je zaměřen na jednoduchou, vysoce optimalizovanou sadu strojových instrukcí, která je v protikladu s množstvím specializovaných instrukcí ostatních architektur (například u RISC neexistuje instrukce pro násobení, která se realizuje softwarově pomocí jednoduchých instrukcí sčítání a bitových posunů). Přesná definice označení RISC není jasná, avšak často se používá přesnější název architektura load-store, která lépe vyjadřuje fakt, že celkový počet instrukcí RISC procesoru může být paradoxně vyšší, než u jiných architektur.[3]

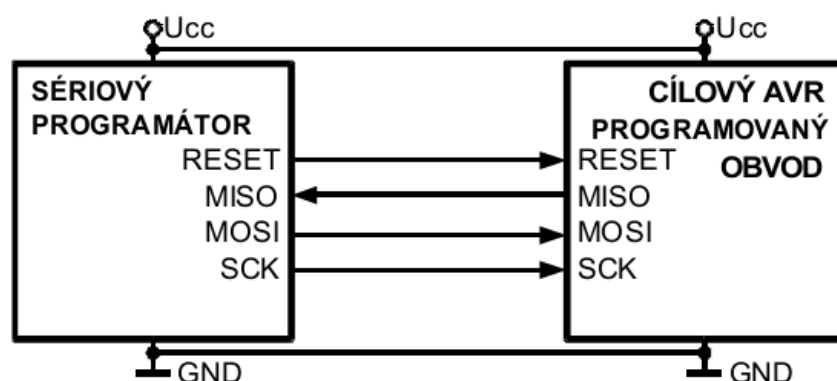
3 Možnosti změny firmware v mikrokontrolérech

V elektronice a výpočetní technice je termín firmware často používán k označení něčeho stálého, většinou poměrně malého. Jedná se o programy nebo datové struktury, které vnitřně ovládají různá elektronická zařízení.[4]

V mikrokontrolérech se jedná o program uložený v programové paměti, která je elektricky mazatelná a programovatelná. Rodina AVR má paměť programu FLASH typu EEPROM. Jedná se o elektricky mazatelnou paměť pro ukládání nevelkého množství dat, přičemž data jsou v paměti zachována, i když je zařízení odpojeno od elektrického napájení.

3.1 Sériové programování

Celou rodinu AVR řady ATmega lze programovat sériově přímo v zapojení, ve kterém se pak bude provozovat aplikace viz obr. č. 3. Sériovému programování pomocí synchronního sériového portu SPI se vžil zkratka ISP. Paměť programu i dat je přístupná po bytech od adresy 0000 až po svou jmenovitou velikost. K vlastnímu programování slouží pouze tři signály SCK, MISO, MOSI.



Obr. 3 - Šestivodičové zapojení mezi programátorem a programovaným mikrokontrolérem

Programovací sekvence je pak následující:

- připojit napájecí napětí a uvést RESET do log. 1 a vyčkat nejméně 10ms,
- na vývod MOSI vyslat instrukci povolující sériový zápis do paměti,
- vysláním příslušně formované instrukce zápisu naprogramovat 1 byte,
- zápis lze verifikovat zasláním instrukce pro čtení, která na vývodu MISO vrátí obsah žádané adresy,
- po poklesu RESET do log. 0 se MCU vrátí k běžné činnosti.

Sériové programování lze zakázat naprogramováním pojistky v paralelním módu. Po zakázání této pojistky není možno s mikrokontrolérem přes SPI komunikovat. To lze také znemožnit při deaktivaci pinu RESET (konfigurační bit RSTDISBL), aby mohl být použit jako chybějící I/O, vypnutí možnosti ISP programování (bit SPIEN).[5][6]

Aby byla možná další komunikace přes SPI, je nutné ji nejdříve povolit a to pomocí paralelního přeprogramování pojistek nebo pomocí JTAG programátoru. Je však nutno podotknout, že lze zakázat i programování přes JTAG rozhraní a jediná šance poté na znovu zprovoznění mikrokontroléru je přeprogramování paralelním programátorem, který využívá HVP programování.

Mikrokontroléry ATmega obsahují dva Lock Bity LB1 a LB2. Jedná se o dva bity, které určují, zda je povoleno z mikrokontroléru číst nebo do něj zapisovat. Nastavení Lock Bitů určuje tabulka č. 1.

Mód	LB1	LB2	Typ ochrany
1	1	1	Zápis ani čtení není nijak omezeno
2	0	1	MCU lze přečíst, avšak programování je zakázáno
3	0	0	MCU nelze přečíst ani do něj nic zapsat

Tab. 1- Lock Bity mikrokontrolérů ATmega

Uzamčení MCU pomocí Lock Bitů slouží především k znemožnění kopírování obsahu paměti z jednoho do více mikrokontrolérů a nedovolenému šíření jeho obsahu. Tyto bity lze vymazat jakoukoliv metodou programování. Příkazem ERASE se smaže obsah paměti mikrokontroléru a následně jde přenastavit i tyto lock bity. Poté lze znovu mikrokontrolér naprogramovat a modifikovat jeho nastavení.

3.2 Paralelní programování

Pokud je mikrokontrolér **mimo aplikaci**, lze jej také naprogramovat pomocí paralelního programátoru. Paralelní programování využívá HVP programování a tedy přivádí na pin RESET vysoké napětí v rozmezí 11,5 – 12,5 V. [6]

Díky zvýšenému napětí na pinu RESET se musí dát pozor, jelikož ne každý mikrokontrolér tuto možnost podporuje a může se vysokým napětím zničit.

Paralelní programování se využívá velmi zřídka, jelikož k přeprogramování je nutno připojit velké množství pinů MCU a také zvýšené napětí by se muselo odpojit od ostatních komponent v aplikaci.

I přes veškeré nevýhody spojující paralelní programování se tato metoda používá pro “znovuoživení” mikrokontrolérů, které byly nepozorností uzamčeny nebo je zapotřebí aktualizovat firmware mikrokontroléru, který má pin RESET použit jako klasický I/O pin.

3.3 JTAG (Joint Test Action Group)

S postupným nárůstem výrobců integrovaných obvodů a jejich neustálému zdokonalování se vyskytla myšlenka nejen tyto obvody programovat, ale také za běhu je moci ladit či provádět různé nastavení.

Vznikla tedy skupina označovaná jako JTAG. Ta usiluje o dosažení dohody, která by stanovila princip konstrukce integrovaných obvodů. Cílem této skupiny je, aby metodika testování integrovaných obvodů byla použitelná i v případě, že se na jedné desce vyskytují komponenty od různých výrobců.

Současná verze této metodiky, reprezentovaná normou IEEE 1149.1, je výsledkem sjednocení názorů výrobců integrovaných obvodů a jejich zákazníků. Je to kompromis mezi snahou o respektování požadavků diagnostiky a snahou o co nejnižší výrobní náklady.

Norma IEEE 1149.1 tedy definuje testovací logické obvody, které mohou být zařazeny do integrovaných obvodů za účelem testování propojení mezi jednotlivými komponentami na již osazených deskách, testování integrovaného obvodu samotného a sledování chování integrovaného obvodu při jeho normální činnosti.

Obvody vyhovující normě JTAG dovolují, aby do jejich komponent byly načteny testovací instrukce a data a po provedení testovacích instrukcí pak provést výstup výsledků testování. Komunikace s JTAG-em (přenos instrukcí, testovacích dat a výsledků testů) probíhá sériově.[7]

Jedná se pouze o standard definující komunikaci, ne však o ucelené zařízení nebo jakoukoliv HW podobu propojování komunikujících periférií. Díky tomuto nedostatku se při návrhu ladicích prostředků různých výrobců setkáváme s odlišnými konektory JTAG adaptérů.

JTAG je možné použít kromě primárního účelu, kterým je testování plošných spojů a interní funkce obvodů, také k programování FLASH pamětí, procesorů, FPGA, CPLD a dalších. [8]

Pro přístup k testovací logice a jejímu řízení slouží tzv. Test Access Port. Je sestaven minimálně ze tří vstupních a jednoho výstupního signálu.

- TCK – vstupní signál, který se používá nezávisle na hodinách jednotlivých komponent. Díky tomu probíhá posuv dat paralelně s činností komponent.
- TMS – vstupní signál je dekodován řadičem TAP a slouží k výběru testovacích operací. Hodnota na vstupu se snímá při náběžné hraně TCK.
- TDI – vstupní signál používající se k přenosu dat. Hodnota na vstupu se snímá při náběžné hraně TCK.
- TDO – výstupní signál, pomocí něž se vysílají sérově požadovaná data z komponenty.
- TRST – nepovinný signál slouží k resetování integrovaného obvodu [7].

JTAG rozhraní mikrokontroléru lze zablokovat (konfigurační bit JTAGEN), lze jej však opět povolit vymazáním konfiguračního bitu pomocí sériového nebo paralelního programátoru.

3.4 Bootloader

Mikrokontroléry řady ATmega jsou schopny používat bootloader k programování paměti FLASH a EEPROM. Paměť FLASH je rozdělena do dvou částí. První část tvoří samotná aplikace a druhou volitelnou částí je bootloader. Velikost jednotlivých částí se nastavuje pomocí konfiguračních bitů

BOOTSZ (Boot Size). Čtení a zápis paměti provádí instrukce LPM (Load Program Memory) a SPM (Store Program Memory), které se vykonají pouze v paměťové části určené pro bootloader.

Standardně se po zapnutí (resetu) obvodu provádějí instrukce od adresy nula. Bootloader je však umístěn na samý vrchol paměti FLASH. Je proto nezbytné nastavit konfigurační bit BOOTRST (Boot Reset), čímž se po startu začne na adrese bootloader-u. Další chování je dáno pouze tím, jak je napsán kód bootloader-u, například pokud splníme určitou podmínku, dojde k aktualizaci firmwaru, v opačném případě se spustí aplikace.[9]

Bootloader není plnohodnotný programátor a tudíž neumožňuje plnohodnotnou manipulaci s konfiguračními a zabezpečovacími bity. V plném rozsahu je lze pouze číst, zápis je omezen jen na několik vybraných bitů. Lze s ním však aktualizovat nejen firmware mikrokontroléru, ale i samotný bootloader.

Výhoda bootloader-u je, že kód, který chceme nahrát do programové paměti, musí být odněkud získán – na rozdíl od hardwarového programování (kde jsou pevně určeny vodiče a protokol) lze použít libovolný prostředek mikrokontroléru (sériová linka, USB, SPI, TWI nebo vlastní rozhraní). Běžící program tedy může například při startu nebo pravidelně za běhu kontrolovat, zda je k dispozici nová verze softwaru, provést její nahrání do programové paměti a spustit ji.[10]

3.5 DebugWIRE

Se zvyšováním nároků na HW možnosti mikrokontrolérů museli výrobci přijít i s dalšími inovacemi. Jedna z vylepšení mikrokontrolérů je možnost ladění i bez přítomnosti JTAG rozhraní. Vzniklo tak nové ladící rozhraní DebugWIRE, které ke své činnosti potřebuje pouze jediný pin. Jedná se o pin, který se používá pro vnější reset.

Pomocí tohoto rozhraní má uživatel plnou kontrolu nad celou pamětí, ze které může číst nebo do ní zapisovat. Také je podporováno ladění single-step, run-to-cursor, step-out, a software break.

DebugWIRE nepodporuje plnohodnotnou manipulaci s konfiguračními a zabezpečovacími bity.

4 Návrh řešení pro vzdálenou aktualizaci firmware

Mikrokontroléry se většinou používají ve vestavěných řídicích systémech. Jedná se o jednoúčelové systémy, ve kterých je řídicí počítač (mikrokontrolér) zcela zabudován do zařízení, které ovládá. Vzhledem k tomu, že systém je určen pro konkrétní účel, mohou tvůrci systém při návrhu optimalizovat pro konkrétní aplikaci, a tak snížit cenu výrobku. Vestavěné systémy jsou často vyráběny sériově ve velkém množství, takže úspora bývá znásobena velkým počtem vyrobených kusů. [11]

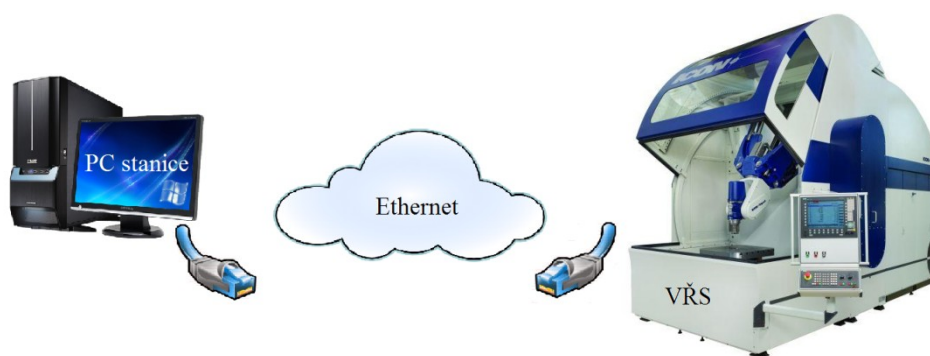
Požadavky na vestavěný řídicí systém se časem zvyšují díky rostoucím nárokům zákazníka a také neustále se zdokonalující technologii. V případě výměny části zařízení za novější typ nebo přidání nových prvků do systému již není schopný řídicí systém tyto prvky ovládat.

Aby celý systém fungoval korektně a s maximální efektivitou, je nutno provést aktualizaci firmware řídicího systému. V případě neaktualizování firmware se může stát, že některé části systému budou vykazovat mnohem horší vlastnosti, než deklaruje výrobce nebo dokonce nebudou fungovat vůbec.

Pokud vezmeme v potaz, že firma vyrábějící zařízení sídlí i několik tisíc kilometrů od zákazníka, který využívá její zařízení, je téměř nemožné jakkoliv urychleně řešit problémy se zařízením osobně servisním technikem firmy.

V době ethernetové konektivity se proto přímo nabízí, zařízení vybavit ethernetovým připojením a případnou aktualizaci firmware nebo vyčtení chyb ze zařízení řešit online na dálku viz obrázek č.4.

Výrobce tak dostává obrovskou výhodu a absolutní kontrolu nad zařízením, kdy nejen může aktualizovat firmware, ale také sledovat chod celého zařízení a případné chyby ihned opravit.



Obr. 4 - Diagram vzdálené aktualizace VŘS pomocí ethernetu [12]

V případě, kdy firemní technik provádí údržbu nebo opravu zařízení přímo na místě, nemusí s sebou nosit žádné drahé a těžké vybavení a případný update firmware může provést pomocí svého mobilního telefonu nebo notebooku bezdrátově pomocí Bluetooth viz obrázek č.5. Bluetooth modul se nejprve spáruje s nadřazeným zařízením (notebook, mobilní telefon...) a následně přes virtuální COM port provede aktualizaci firmware.



Obr. 5 - Diagram vzdálené aktualizace VŘS pomocí Bluetooth [12]

Abychom se při návrhu připojení systému do sítě ethernet vyhnuli složitému programování a nastavování všech vrstev a rozhraní potřebných pro komunikaci, zvolili jsme již hotový výrobek od firmy Connect One modul NanoSocketLAN, který slouží pro připojení zařízení do sítě 10/100BaseT Ethernet LAN.

4.1 Ethernetový modul NanoSocketLAN

Miniaturní modul NanoSocketLAN (obrázek č.6) od firmy ConnectOne je určen ke snadnému připojení jakékoli aplikace do sítě 10/100BaseT Ethernet LAN.

NanoSocketLAN dovoluje nasazení LAN konektivity bez složitého programování TCP/IP stacku nebo potřeby ovladače pro mikrokontrolér. K mikrokontroléru AVR lze připojit pomocí rozhraní SPI nebo UART. V tomto projektu bylo zvoleno rozhraní UART. Sofistikovaný AT+i API od Connect One eliminuje potřebu jakýchkoli LAN ovladačů, bezpečnostních nebo síťových protokolů nebo jiných úloh pro mikrokontrolér. [13]



Obr. 6 - Ethernetový modul NanoSocketLAN [14]

4.1.1 Podporované protokoly

Modul NanoSocketLAN podporuje velké množství internetových protokolů, mezi které patří například ARP (protokol sloužící k zjištění MAC adresy zařízení v lokální síti, když známe jeho IP adresu), ICMP (jeden z nejdůležitějších protokolů, používají ho operační systémy v síti pro odesílání chybových zpráv), FTP (komunikační protokol používaný hlavně pro přenos souborů), DHCP (Používá se pro automatickou konfiguraci počítačů připojených do počítačové sítě. DHCP server přiděluje počítačům pomocí DHCP protokolu zejména IP adresu, masku sítě, implicitní bránu a adresu DNS serveru).

Ke své činnosti používá také bezpečnostní protokoly například HTTPS (Jedná se o nádstavbu síťového protokolu HTTP, který umožňuje zabezpečit spojení mezi webovým prohlížečem a webovým serverem před odposloucháváním, podvržením dat a umožňuje též ověřit identitu protistrany), FTPS (jde o rozšíření protokolu FTP, ve kterém je veškerá komunikace šifrovaná), 3DES (jedná se o silný šifrovací algoritmus).

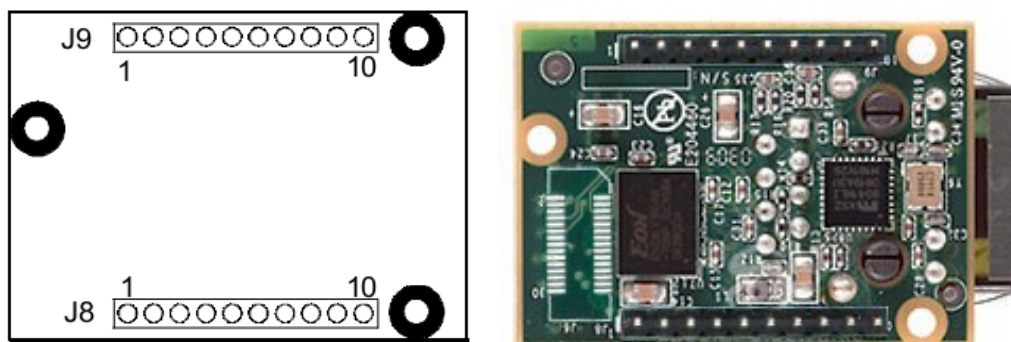
Modul má také akcelerované HW protokoly AES, 3DES a SHA

4.1.2 Přenosové rychlosti

Při komunikaci v sériovém módu dosahuje modul rychlosti přenosu dat až 3Mbps/3Mbps. Nejrychleji odesílá a přijímá data v SPI módu, kdy se rychlost dostane až na 24Mbps / 5Mbps. USB mód poskytuje přenosové rychlosti 12Mbps / 500Kbps.

4.1.3 Schéma zapojení vývodů

Pro připojení k aplikaci slouží ze spodní strany modulu dvě jednořadé, 10-pinové lišty s roztečí 2mm. Modul lze dodatečně osadit i konektorem Molex 2*15 pinů nebo jej rovnou v této variantě zakoupit jako modul Nano LANReach. Zapojení vývodů přehledně ukazuje obrázek č.7.



Obr. 7 - Zapojení vývodů modulu NanoSocketLAN [15]

Význam jednotlivých vývodů konektorů J8 a J9 je v následujících tabulkách č.2 a č.3.

Pin	Signál	Typ	Popis
1	GND	napájen	zem
2	VDD	napájen	+ 3,3 V (+/- 10%)
3	RXD0	vstup	UART 0 příjem
4	TXD0	výstup	UART 0 vysílání
5	nCTS0	vstup	UART 0 (clear to send)
6	nRTS0	výstup	UART 0 (request to send)
7	DATA_RD	výstup	data připravena
8	MSEL	vstup	výběr módu (záchranný režim / aktualizace firmware / uživatelský
9	nRESET	vstup	reset modulu NanoSocketLAN
10	ACT_LINK	výstup	LED indikátor

Tab. 2- Význam vývodů konektoru J8 modulu NanoSocketLAN

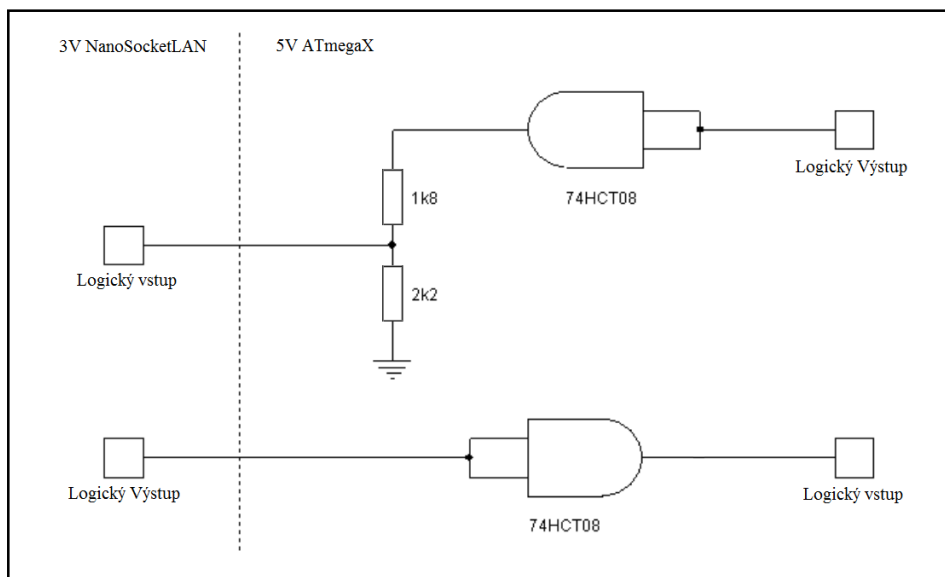
Pin	Signál	Typ	Popis
1	nSPI1_CS	vstup	SPI1 CS
2	nSPI1_CLK	vstup	SPI1 hodinový signál (max 12 MHz)
3	SPI1_MISO	výstup	SPI1 MISO
4	SPI1_MOSI	vstup	SPI1 MOSI
5	SPI1_INT	výstup	SPI1 plný buffer
6	readiness	výstup	Ichip připraven
7	DDM	analog	USB zařízení
8	DDP	analog	USB zařízení
9	SPEED	výstup	LED indikátor 0 = 100Mbit/s, 1 = 10Mbit/s
10	GND	napájení	zem

Tab. 3- Význam vývodů konektoru J9 modulu NanoSocketLAN

4.1.4 UART rozhraní

Komunikace modulu NanoSocketLAN s mikrokontrolérem ATmega bude probíhat pomocí rozhraní UART. Z důvodu rozdílného pracovního napětí, kdy modul NanoSocketLAN používá napěťové

úrovně 3,3V a mikrokontrolér ATmega 5V, je zde zařazen napěťový převodník 5V/3,3V složený ze dvou hradel a dvou rezistorů, viz obrázek č. 8.



Obr. 8 - Napěťový převodník 5V / 3,3V

4.2 JTAG programátor

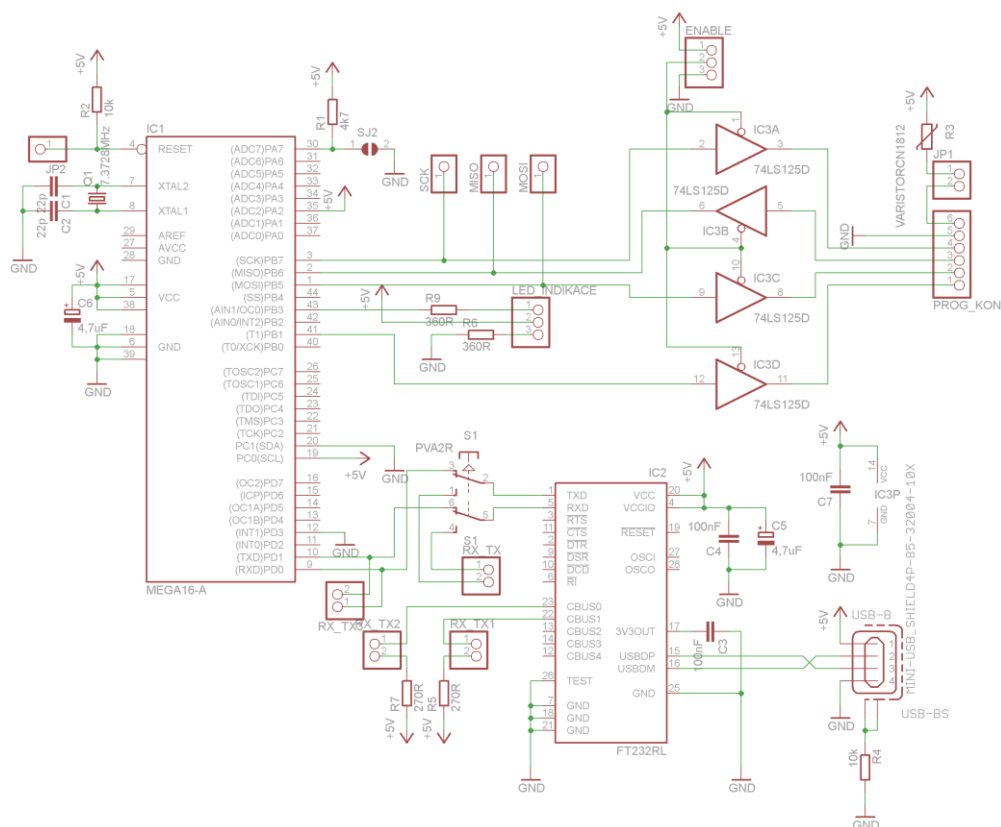
Z kapitoly č. 3 Možnosti změny firmware v mikrokontrolérech jasně plyne, že JTAG programátor má obrovskou výhodu oproti jiným programátorům v možnosti také ladit aplikaci za běhu, vyčítat hodnoty registrů, atd.

Díky této skutečnosti bude na vývojovou desku umístěn JTAG programátor a pomocí něj se bude firmware mikrokontrolérů měnit.

Nejprve je nutno JTAG programátor navrhnout a otestovat, aby při pozdější implementaci do vývojové desky nedošlo k situaci nefunkční části a tedy k nemožnosti aktualizace firmware.

4.2.1 Schéma JTAG programátoru

Schéma JTAG programátoru (obrázek č.9) se skládá ze dvou částí. První částí je převodník USB-UART od firmy FTDI čip FT232. Druhou částí je mikrokontrolér ATmega16 od firmy ATMEL. Čipy se mezi sebou propojí sériovou linkou. Výstup JTAG programátoru je na pinech PB7 (TCK), PB6 (TDO), PB5 (TDI) a PB1 (TMS). Indikační LED dioda je připojena na port PB3.

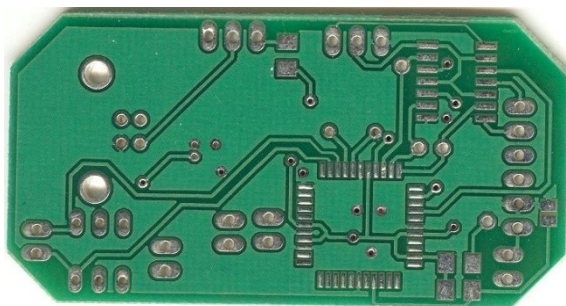


Obr. 9 - Schéma JTAG programátoru

Jedná se o klon JTAG programátoru od firmy ATMEL AVR JTAG ICE. Podrobnějším nastavením, zavedením firmware a následným update firmware se zabývá kapitola 5.2.4 Aktualizace firmware JTAG programátoru.

4.2.2 DPS JTAG programátoru

Deska plošného spoje má rozměry 59,5 mm x 30,5 mm x 1,5 mm. Použitý materiál FR4 s nanesenou nepájivou zelenou maskou. Měděné pájitelné plošky jsou ošetřeny metodou HAL. DPS a osazovací list se nachází v příloze I a II. Vyrobenou DPS zobrazuje obrázek č.10.



Obr. 10 - JTAG programátor

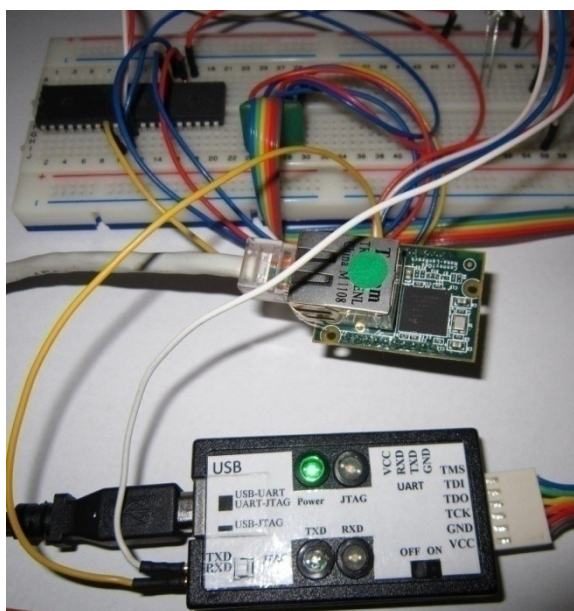
Tento rozměr je vnitřním rozměrem standardní krabičky U-KM57A. Krabička se skládá ze dvou dílů, které do sebe zapadají. Na boku má hranatý otvor o rozměrech 6 mm x 13 mm.

Indikační LED diody, přepínač signálu, USB port a konektory s výstupem a vstupem signálu UART nemají v krabičce díry. Plast je velice snadno opracovatelný a je nutno tyto otvory do něj vyvrtat a vyřezat. Po osazení DPS všemi součástkami a vyvrtání otvorů do krabičky je vhodné nalepit na programátor popis jednotlivých vstupů, výstupů, indikačních LED diod, atd... Vyrobený JTAG programátor ukazuje obrázek č.11.



Obr. 11 - JTAG programátor

Po otestování samotného JTAG programátoru připojeného přes USB je nutno odzkoušet propojení a funkčnost i s modulem NanoSocketLAN pomocí sériové linky viz obrázek č.12.



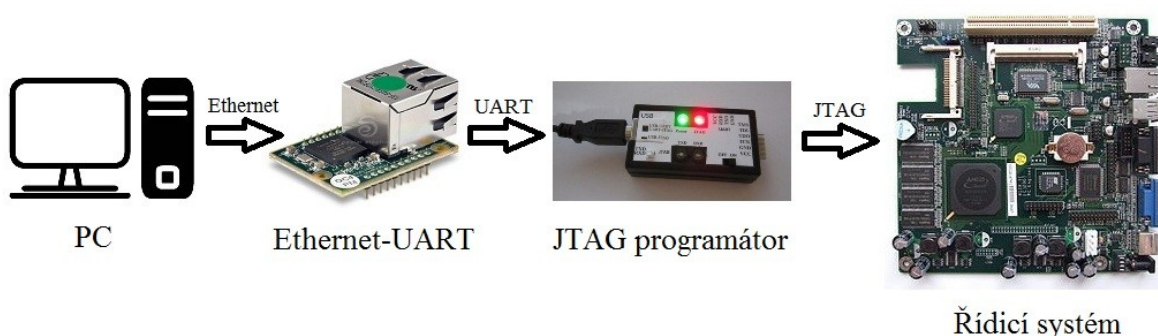
Obr. 12 - Propojení JTAG programátoru s modulem NanoSocketLAN

Po propojení JTAG programátoru s modulem NanoSocketLAN je nutné nejprve nastavit tento modul a komunikaci po ethernetu, která je popsána v kapitole 5. Po úspěšném propojení modulu s programátorem spustíme vývojové prostředí AVR studio 4 a otestujeme, zda lze naprogramovat mikrokontrolér a také jej ladit.

5 Vzdálená aktualizace firmware pomocí PC

Vzdálená aktualizace firmware začíná instalací nezbytného software a jeho správného nastavení. Pomocí AVR studia zkontrolujeme nejnovější firmware JTAG programátoru a popřípadě provedeme update. O překlad kódu z jazyka C, naprogramování mikrokontroléru i samotné krokování se stará taktéž AVR studio. Blokové schéma jednotlivých prvků vzdálené aktualizace ukazuje obrázek č. 13.

Vývojová stanice (PC, notebook, atd.) se pomocí ethernetu připojí k modulu NanoSocketLAN. Tento modul je připojen k JTAG programátoru po sériové lince UART. JTAG programátor je připojen k mikrokontroléru pomocí rozhraní JTAG.



Obr. 13 - Blokové schéma vzdálené aktualizace firmware[14]

Před samotnou vzdálenou aktualizací je nejprve nutné provést řadu nastavení HW i SW, bez kterého by ji nebylo možné provést.

5.1 Instalace software

Aby bylo možné nastavit modul NanoSocketLAN, je zapotřebí nainstalovat software iChipConfig. Pro vytvoření virtuálního portu program HW VSP3. Zkompilování zdrojového kódu a samotné programování zajišťuje AVR studio. A pokud budeme používat převodník USB-UART, je nutné také doinstalovat ovladače k převodníku FT232. Instalace nezbytného software je popsána v příloze v kapitole XI. Instalace nezbytného software.

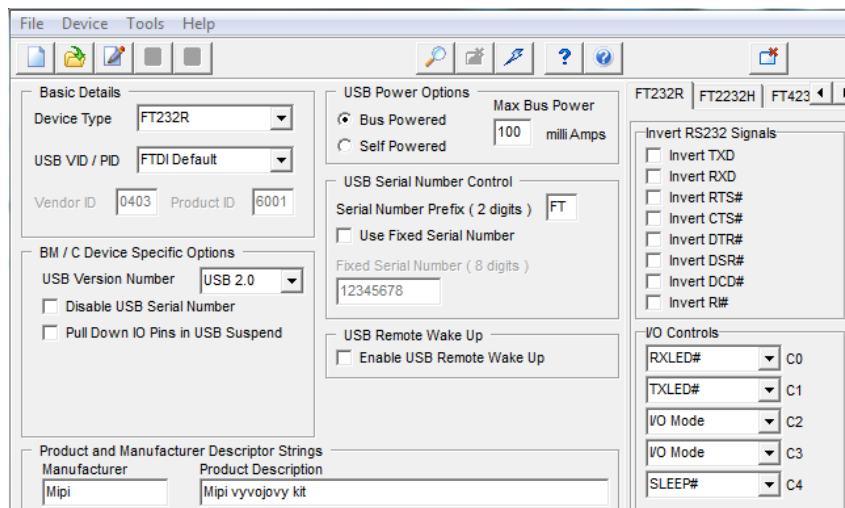
5.2 Nastavení SW a HW

Modul NanoSocketLAN lze nastavit pouze po sériové lince s napětovými hodnotami 3,3V nebo pomocí virtuálního TCP/IP COM portu umožňujícího přenos sériových dat přes virtuální sériový port, který se chová jako běžný hardwarový COM port.

5.2.1 Nastavení FT232

Převodník USB-UART FT232 má nějaké základní nastavení, které však nemusí korespondovat s našimi požadavky. Toto nastavení se mění pomocí programu MProg (obrázek č.14). Po spuštění

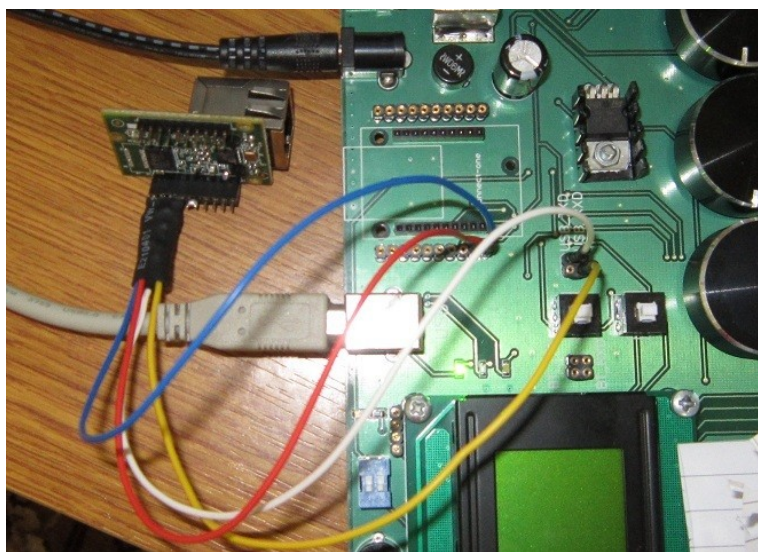
vybereme z roletové nabídky File možnost Open a z příloženého CD otevřeme ept soubor s názvem Mipi_ft232.ept. Tento soubor obsahuje přednastavené hodnoty, které se pomocí příkazu Program all existing devices zapíše do EEPROM paměti převodníku. Takto nastavený převodník se po připojení k PC zahlásí jako Mipi vyvojovy kit a na pinech C0 a C1 je nastavena indikace činnosti RXD (bílá LED) a TXD (modrá LED).



Obr. 14 - Nastavení EEPROM v programu MProg

5.2.2 Nastavení NanoSocketLAN

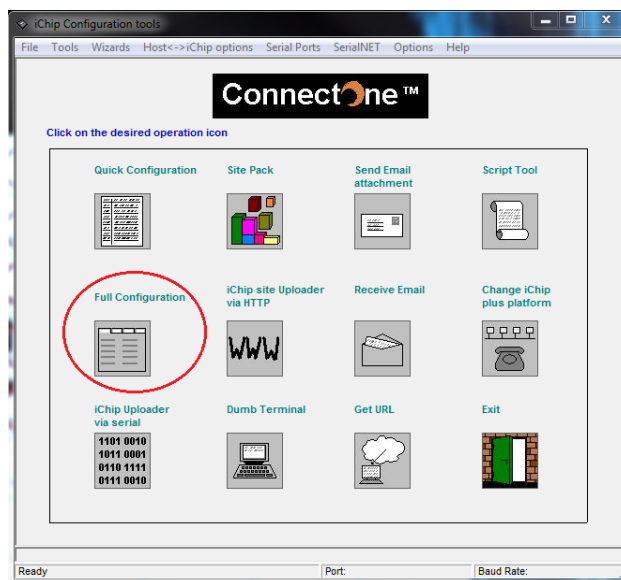
Modul NanoSocketLAN před prvním použitím je nutno vždy vytáhnout z vývojové desky tak jak ukazuje obrázek č.15 a nastavit. První nastavení se provádí pomocí příložené redukce. Ta obsahuje 4 vodiče (modrá - GND, červená - +3,3V, bílá - RXD, žlutá - TXD). Po nakonfigurování modulu, pokud se nepoužívají, je nutné spojit piny nCTS0 a nRTS0, jinak nebude komunikace fungovat.



Obr. 15 - První nastavení modulu NanoSocketLAN

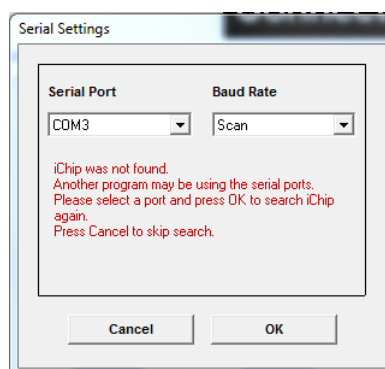
K vývojové desce připojíme USB kabel, napájení a vodiče z redukce (modrá - GND, červená - +3,3V, bílá - USB_TXD, žlutá – USB_RXD).

Spustíme program iChipConfig a klikneme na ikonu Full Configuration. Na obrázku č. 16 je zvýrazněna červeně.



Obr. 16 - program iChip Configuration tools

Pokud se objeví okno Serial Settings (obrázek č. 17), nemůže program iChipConfig najít modul NanoSocketLAN a je zapotřebí ručně vybrat COM port a přenosovou rychlost Baud Rate. V hardwareové konfiguraci vývojového kitu zjistíme, na jakém COM portu zařízení komunikuje. Do pole Serial Port zadáme používaný COM a do pole Baud Rate zadáme scan. Nyní program iChipConfig proskenuje všechny přenosové rychlosti na zadaném COM portu a připojí se na stejné rychlosti jakou modul NanoSocketLAN komunikuje.



Obr. 17 - Výběr sériového portu a přenosové rychlosti

Po úspěšném připojení se objeví okno s nastavením modulu.

- záložka Operational:

Host Interface (HIF) – 1 - UART 0

- záložka LAN

Default IP address (DIP) – zadat požadovanou IP adresu (192.168.0.150)

Sub Net address (SNET) – zadat masku podsítě (255.255.255.0)

Gateway IP address – zadat adresu gateway (192.168.0.1)

V případě použití režimu DHCP klient zaškrtnout políčko Use DHCP

- záložka Serial NET

Server serialNET port (LPRT) – zadat požadovaný port (5003)

IP address to connect to (HSRV) – zadat požadovanou IP adresu a požadovaný port (192.168.0.150:5003)

Port settings for serialNET (SNSI) – zadat parametry sériové linky ve tvaru (9,8,N,1,0).

Význam jednotlivých znaků v položce SNSI vysvětluje tabulka č.4.

Význam jednotlivých znaků:				
Baudrate	Data bity	Parita	Stop bity	Řízení dat
1=600	7 = 7 bitů	N = bez parity	1 = 1 stop bit	0 = bez řízení
2=1200	8 = 8 bitů	E = sudá (Even)	1.5 = 1.5 stop bitu	1 = HW flow
3=2400		O = lichá (Odd)	2 = 2 stop bity	
4=4800				
5=9600				
6=19200				
7=38400				
8=57600				
9=115200				
h=230400				

Tab. 4- Význam jednotlivých znaků v SNSI

iChip Configuration tools

File Search Parameter

ISP Connection	Remote params update	HTTP	Serial NET	IP registration
User fields	RAS	Wireless LAN	SSL	Router
Email format	Server profiles	Operational	LAN	Remote FW

Extended return code (XRC)
☐ 4 - No Blind Dial
☐ Blind dial

Modem initialization string (MIS)
 AT&F&OY1X4Q&D2M1L3

Modem type designator (MTYP)
☐ Standard Modem

☐ Suppress ATZ

Wait time constant (WTC)
 45

TCP timeout (TTO)
 0

Inactivity timeout (IATO)
 0

CDPD Password (CPWD)

Max PPP Packet Size (MPS)
☐ 0 - Auto

Host Interface (HIF)
☐ 1 - UART 0

A2D detect period interval (ADCT)
 0

Host wakeup min interval (S100)

Host wakeup delay time (S102)

Bridge Mode (BRM)
☐ 0 - Disable

Baud rate (BDRM)
☐ a - Auto Baud Rate

Fix baud rate (BDRF)
☐ a - Auto Baud Rate

Flow control (FLW)
☐ No flow control

☐ Disable reflection

Dial Mode (DMD)
☐ Tone

Activate Web Server (AWS)
☐ Disabled

Triggered UP (TUP)
☐ Disabled

Modem Interface (MIF)
☐ 2 - UART 1

A2D detect threshold level (ADCL)
 0

A2D detect GPIO Pin
 0

A2D detect threshold delta (ADCD)
 0

SPI Control Signal (SPIP)
 1

MAC Filter (MACF)

☐ Network Time Of Day Active (NTOD)

Network Time Server (NTS1)

Network Time Server 2 (NTS2)
 0

Greenwich Mean Time Offset (GMTO)

Daylight Saving Time Definition (DSTD)

☐ IP Prot. Don't Fragment Bit (DF)

AT+ Socket Server (LATI)
 0

Host name (HSTN)

Baud Rate Divider (BDRD)
 0

Framing Mode (FMODE)
☐ 0 - Framing Disabled

PPP ACFC Negotiation (PPP)
☐ 0 - Agree to ACFC

LATI Restriction (LAR)
 0

IO Configuration (IOC)

☐ Auto Router Start (ARS)

☐ Calculate Check Sum (CKSM)

Timeout to resend PING request (PGT)
 0

Timeout to resend un acked TCP packet (TTR)
 3000

Select PIN 44 function (PIN44)
☐ Clock out

Watch Dog Activation Mode (WDM)

Readiness Report HW (RRHW)
 2

Content Type (CTT)

LAN Type (LTYP)
☐ 0 - Automatic

Web Site Logo file (LOGO)
 iChipimages/ConnectOne.gif

Power Save Mode Enable (PSE)
 0

Service Disable Mask (SDM)
☐ Ping Reply
☐ Remote Debug
☐ Unauth. WEB Server
☐ Disable SNET escape by +++
☐ Disable Break Reset
☐ Disable Internal Config Site

Save Apply Close

Ready Port: COM3 Baud Rate: 115200

Obr. 18 - Nastavení záložky Operational

iChip Configuration tools

File Search Parameter

ISP Connection	Remote params update	HTTP	Serial NET	IP registration
User fields	RAS	Wireless LAN	SSL	Router
Email format	Server profiles	Operational	LAN	Remote FW

MAC address (MACA)
 000394199153

Assigned IP address (IPA)
 192.168.0.150

Default IP address (DIP)
 192.168.0.150

☐ Use DHCP
☐ Use IP Finder

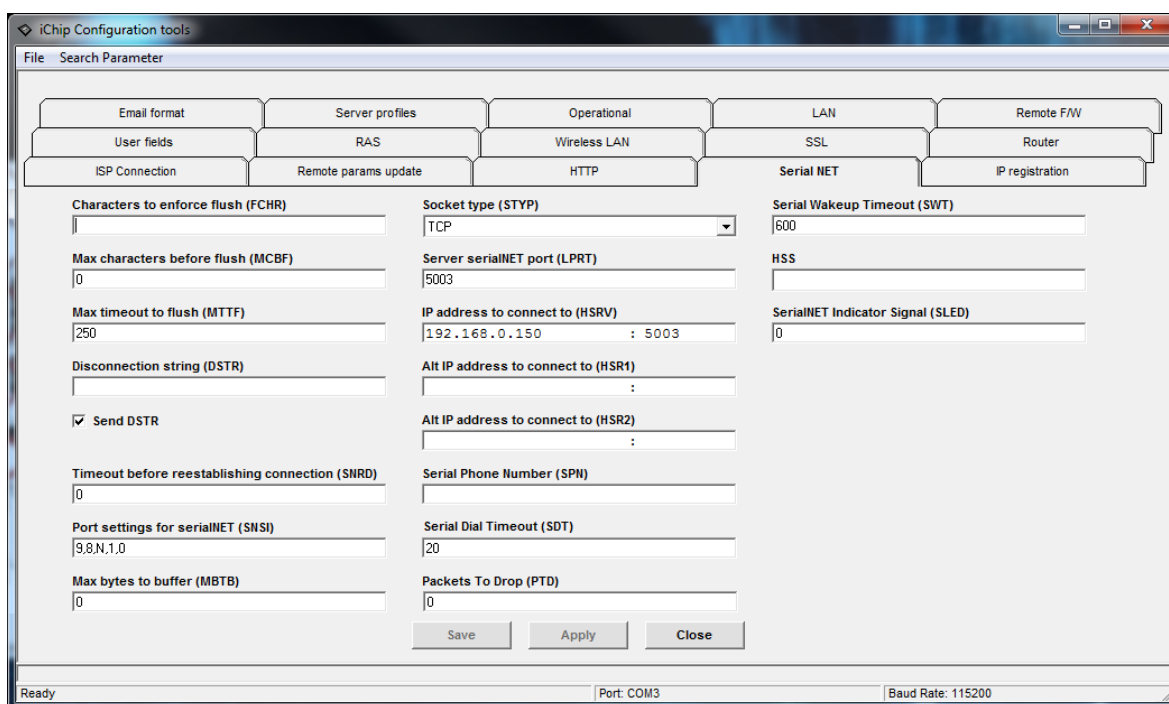
Sub Net address (SNET)
 255.255.255.0

Gateway IP address
 192.168.0.1

Save Apply Close

Ready Port: COM3 Baud Rate: 115200

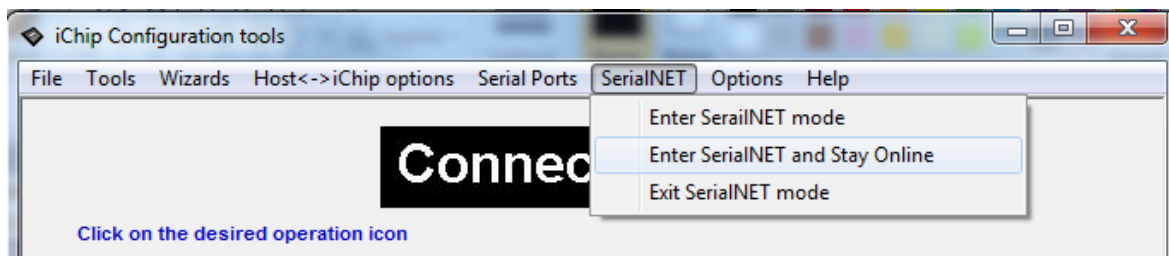
Obr. 19 - Nastavení záložky LAN



Obr. 20 - Nastavení záložky Serial NET

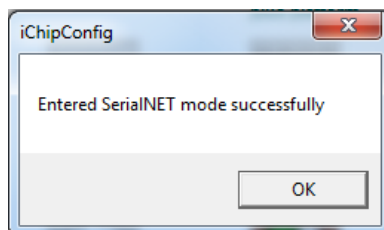
Tímto je nastavení modulu NanoSocketLAN hotovo, takže je nutné jej uložit tlačítkem Save.

Z roletové nabídky SerialNET vybereme příkaz Enter SerialNET and Stay Online. Použitím tohoto příkazu je zajištěno, že se modul po zapnutí napájení bude nacházet v režimu SerialNET.



Obr. 21 - Nastavení modulu do režimu SerialNET

Po nastavení se zobrazí hlášení, že modul byl úspěšně nastaven do módu SerialNET.

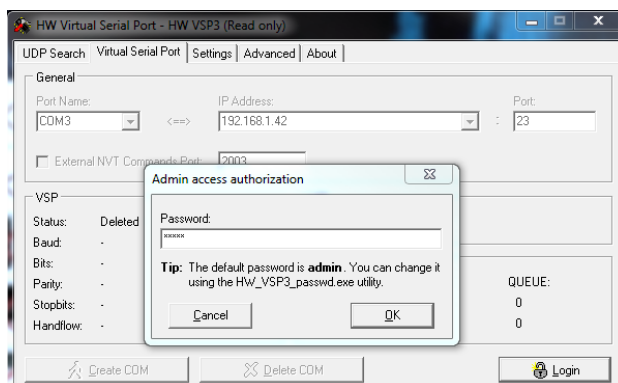


Obr. 22 - Hlášení o úspěšném nastavení do módu SerialNET

5.2.3 Nastavení HW VSP3 (Virtual Serial Port)

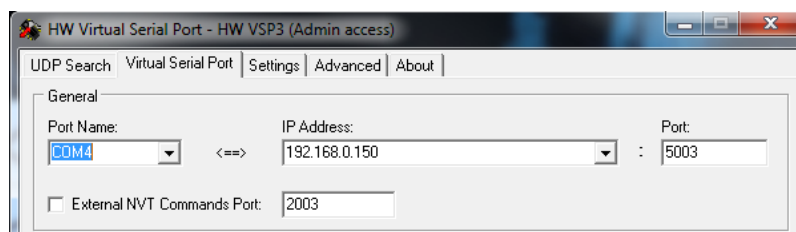
Programů pro vytvoření virtuálního TCP/IP portu je mnoho, většina však komerčních. Existuje několik freeware nebo Open Source možností, poměrně dobrou alternativou je dále popsany český program HW VSP3 od HW Group.[17]

Po spuštění programu klikneme na tlačítko Login a v okně Admin access authorization do políčka Password napíšeme heslo admin viz obrázek č. 23.



Obr. 23 - Zadání hesla

Záložka Virtual Serial Port. Vybereme volný nepoužívaný COM port (COM4), do pole IP Address zadáme IP adresu, kterou jsme zadali do modulu NanoSocketLAN (192.168.0.150) a Port (5003) viz obrázek č. 24.



Obr. 24 - Nastavení záložky Virtual Serial Port

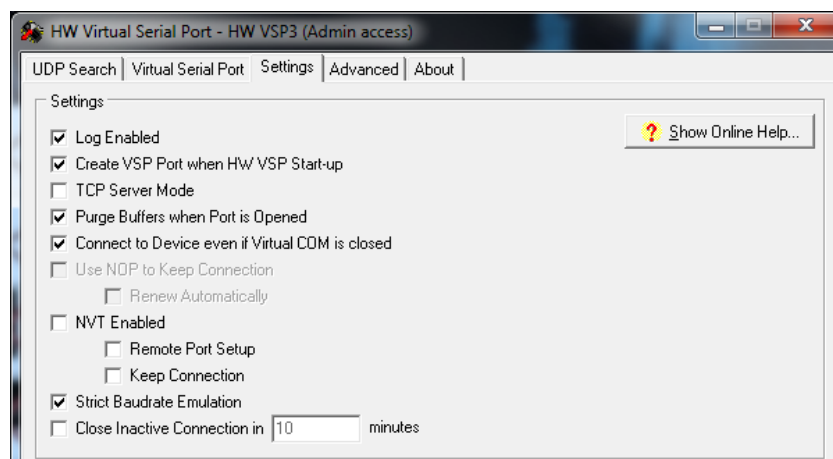
Na záložce Settings zaškrtneme:

- Log Enabled
Loguje veškerou komunikaci do *.log souboru.
- Create VSP Port when HW VSP Start-up
Vytvoří se při spuštění VSP automaticky všechny virtuální porty.
- Purge Buffers when Port is Opened
Při otevření portu vymaže vysílací i přijímací buffer. Tím lze zajistit, že po otevření portu nebudou přijata či odeslána zbytková data z minulé relace, ale současně mohou být vymazána data přijatá před otevřením portu (například při spouštění PC).
- Connect to Device even if Virtual COM is closed

Pokud není zaškrtnuto toto políčko, dojde ihned po vytvoření virtuálního portu k navázání spojení a to i v případě, že tento port není používán žádnou aplikací. To znamená, že budou ztracena data vyslaná v době, kdy byl port otevřen, ale nebyl zatím využíván. Tedy zařízení data posílá i v případě, že je "nepřijímáme".

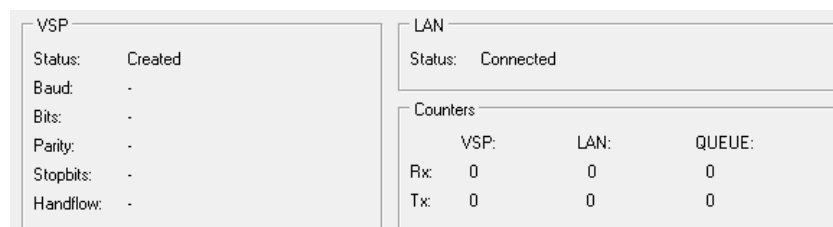
- Strict Baudrate Emulation

Omezí přenosovou rychlost mezi aplikací a VSP a mezi VSP a TCP podle rychlosti otevřeného sériového portu.



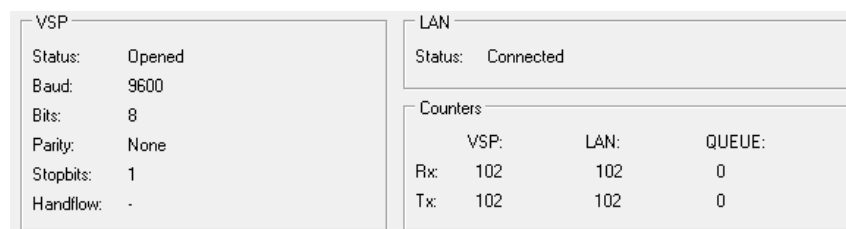
Obr. 25 - Nastavení záložky Settings

Nyní je nastavení dokončeno a na záložce Virtual Serial Port klikneme na tlačítko Create COM. V rámečku VSP se změní status na Created a v rámečku LAN se změní status na Connected.



Obr. 26 - Úspěšné vytvoření VSP

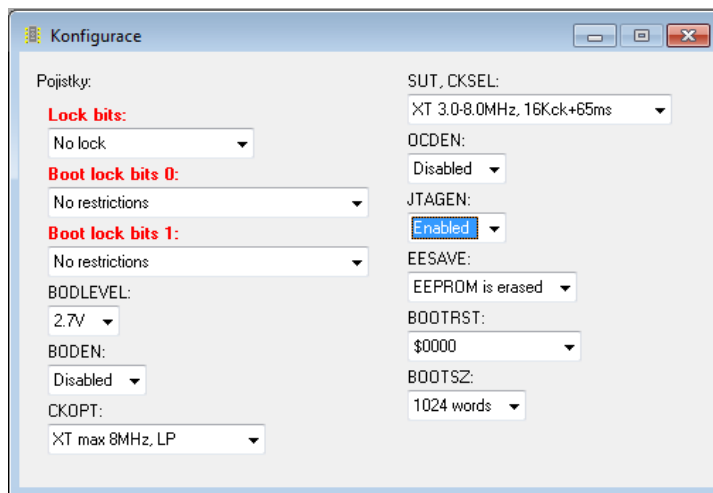
Po spuštění aplikace využívající tento virtuální sériový port se v rámečku VSP zobrazí informace o rychlosti komunikace počtu přenášených bitů paritě a stop bitů. V poli Counters se zobrazuje počet přijatých a odeslaných bitů viz obrázek č.27.



Obr. 27 - Informace o VSP a přehled přijatých a odeslaných bitů

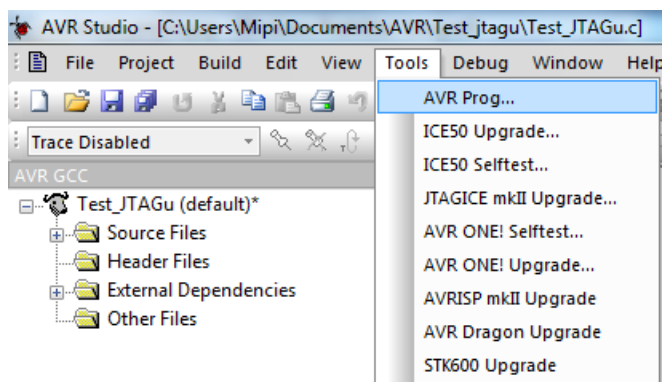
5.2.4 Aktualizace firmware JTAG programátoru

Sestavený JTAG programátor na desce je stejný jako v kapitole 4.2. Nejprve je nutno nahrát firmware do mikropočítače ATmega16. Na desce je vyveden 4 pinový konektor pro ISP programování. Pomocí tohoto konektoru a ISP programátoru nahrajeme firmware do mikropočítače. Jedná se o jakýkoliv bootloader, pomocí kterého aktualizujeme firmware JTAG programátoru. Nastavení pojistek MCU ukazuje obrázek č. 28.



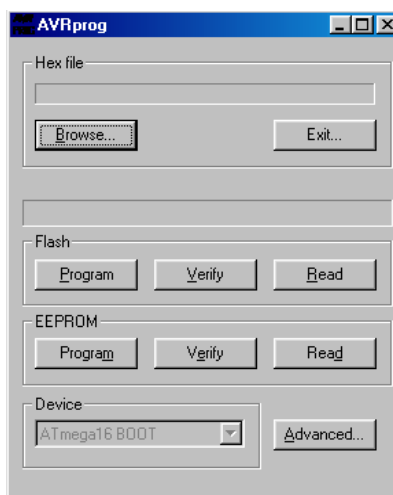
Obr. 28 - Nastavení pojistek mikrokontroléru

Spustíme AVR studio a v roletové nabídce Tools vybereme položku AVR Prog viz obrázek č.29.



Obr. 29 - Výběr položky AVR Prog...

V nově otevřeném okně klikneme na tlačítko Browse a vybereme soubor upgrade.ebn, který se nachází v AVR Studiu defaultně ..\Atmel\AVR Tools\JTAGICE\Upgrade.ebn. Následně provedeme upgrade firmwareu pomocí tlačítka Program v okně Flash. Nyní je JTAG programátor připraven k použití. Programátor JTAG ICE a jeho klony podporuje pouze AVR studio 4 a nižší.



Obr. 30 - Okno sloužící ke změně firmware

5.2.5 Nastavení AVR Studia

Spustíme AVR Studio a v úvodní obrazovce vybereme New Project. Project type vybereme AVR GCC a do pole Project name vepíšeme jméno našeho projektu.

Dalším krokem je výběr Debugovací platformy a mikrokontrolér, se kterým budeme pracovat. Námi vytvořený JTAG programátor je klon JTAG ICE a programovaná součástka je nyní ATmega32. Port ponecháme Auto, avšak námi předem vytvořený virtuální COM port musí být v rozsahu COM1-4, jinak jej AVR Studio nebude detekovat.

Založení nového projektu dokončíme stiskem tlačítka Finish.

5.2.6 Typická ukázka práce s vytvořeným hardware

Do nově vytvořeného okna napíšeme kód v jazyce C a poté následně zkompilujeme stiskem tlačítka Build Active Configuration (1). AVR Studio nám vygeneruje hex soubor, který následně můžeme nahrát pomocí ISP programátoru do mikrokontroléru.

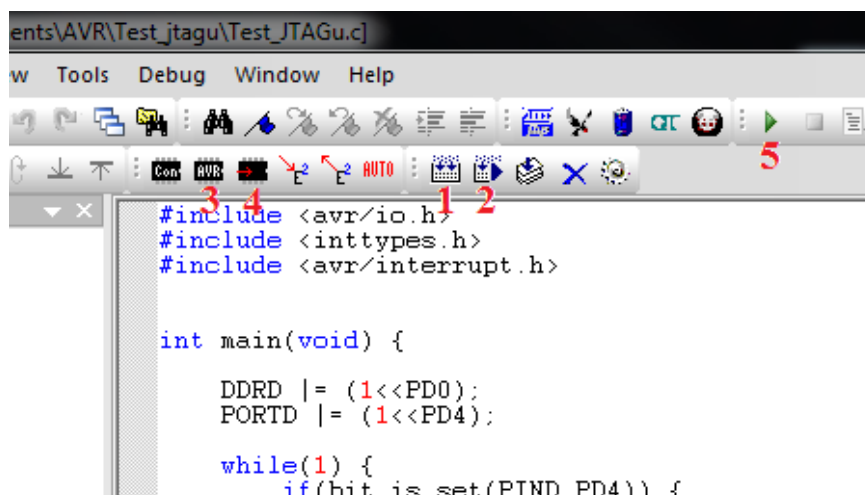
Stiskem tlačítka Build and Run (2) zkompilujeme taktéž náš projekt do hex souboru, ale zároveň pomocí JTAG programátoru jej umístíme do mikrokontroléru a spustíme krokování programu tzv. debugging. V tomto módu lze z mikrokontroléru vyčítat hodnoty registrů, zapisovat do nich, atd.

Pokud chceme pouze mikrokontrolér pomocí JTAGu naprogramovat stiskneme tlačítko AVR (3). Zobrazí se nám okno s nastavením veškerého programování. Vybereme námi požadovaný mikrokontrolér, programovací mód, zdrojový HEX soubor pro paměť Flash, zdrojový soubor pro paměť EEPROM, nastavíme pojistky a lock bity.

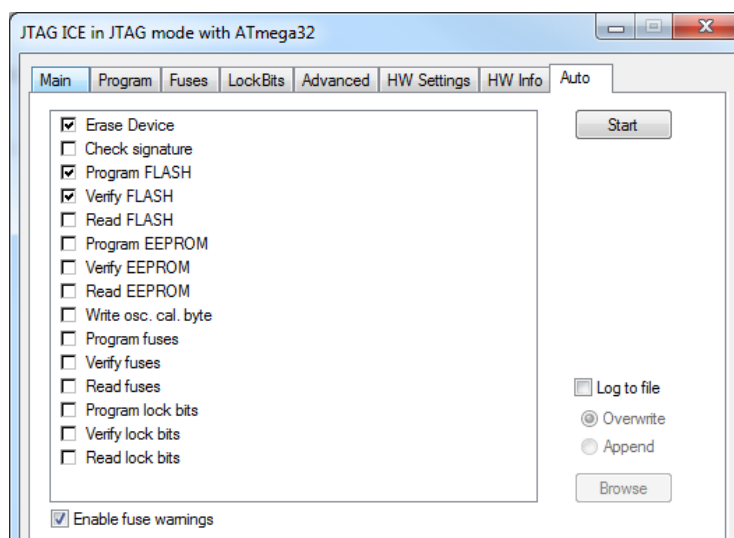
Na záložce Auto nastavíme cyklus programování a stiskneme tlačítko Start. Pokud budeme program měnit a budeme rychle chtít naprogramovat námi definovanou součástku, stačí okno

minimalizovat a poté vždy stisknout pouze tlačítko Write Flash Memory Using Current Settings (4), které se nachází napravo od tlačítka AVR.

Zahájení následného debugingu provedeme tlačítkem Start Debugging (5).



Obr. 31 - Menu kompilace projektu a následného nahrání do mikrokontroléru



Obr. 32 - Kompletní nastavení při programování přes JTAG

6 Návrh vývojové desky

Pokud vývojový pracovník navrhuje nový řídicí systém, je většinou odkázán na nepájivé pole nebo univerzální vývojové desky, kde každou část musí mezi sebou propojit vodiči. Takové univerzální zařízení sice zjednoduší práci, avšak problém s velkým množstvím kabelů a i tím následných poruch v HW nijak neřeší.

Vznikl tedy požadavek vytvořit vývojovou desku, která sjednotí základní typy používaných periférií s nejčastěji používanými typy mikrokontrolérů od firmy Atmel. Vývojová deska musí také obsahovat programátor mikrokontrolérů, který umožní vzdálené přeprogramování.

6.1 Požadavky na vybavení vývojové desky

Vývojová deska musí obsahovat minimálně dva různé mikrokontroléry AVR od firmy Atmel, které půjdou pomocí JTAG programátoru vzdáleně přeprogramovat. LCD displej pro zobrazování informací a dostatečně velké množství přepínačů a tlačítek.

Vývojová deska dále musí obsahovat:

- mikrokontrolér ATmega8 v pouzdře TQFP,
- mikrokontrolér ATmega32 v pouzdře DIL,
- mikrokontrolér ATmega128 v pouzdře TQFP,
- JTAG programátor,
- převodník Ethernet-UART,
- převodník USB-UART,
- bluetooth modul,
- možnost napájení z baterie a po připojení napájecího adaptéru její samočinné odpojení,
- stabilizátor 5V,
- stabilizátor 3,3V,
- indikační LED diody signalizující že je napětí přítomno,
- ISP konektor se zámkem pro naprogramování ATmegy8,
- resetovací obvod s tlačítky pro všechny mikrokontroléry,
- hodinový krystal 32,768kHz,
- patice pro výměnné krystaly,
- LCD displej s možností zapnutí podsvícení,
- bezzákmitové tlačítka,
- bezzákmitové přepínače,
- LED diody,
- TSOP přijímač,
- rotační enkodér s tlačítkem,
- RGB dioda,

- Sirénka,
- Potenciometry,
- převodník UART-RS485,
- nepájivé pole,
- logická sonda,
- EEPROM paměť komunikující po I2C sběrnici,
- konektor pro připojení teplotního čidla Dallas 18B20,
- konektor pro připojení serva,
- konektor s napětím 5V a 3,3V,
- konektor pro připojení externích modulů,
- napěťový dělič pro snímání napětí na baterii,
- veškeré vstupně výstupní piny mikrokontrolérů musí být vyvedeny na dutinkovou lištu,
- veškeré periferie se musí připojovat pomocí přepínačů.

6.2 Požadavky na výrobu vývojové desky

Aby bylo možné desku vyrobit, musí se respektovat určitá pravidla pro výrobu DPS. Při dodržení těchto pravidel je nejen deska vyrobitelná, ale také se předejde problémům, které při výrobě DPS nastávají. Kritéria při návrhu DPS:

- Minimální tloušťka cest 0,2 mm,
- Minimální izolační mezera 0,2 mm,
- Minimální tloušťka potisku 0,15 mm,
- Minimální průměr vrtáku 0,4 mm.

6.3 Schéma vývojové desky

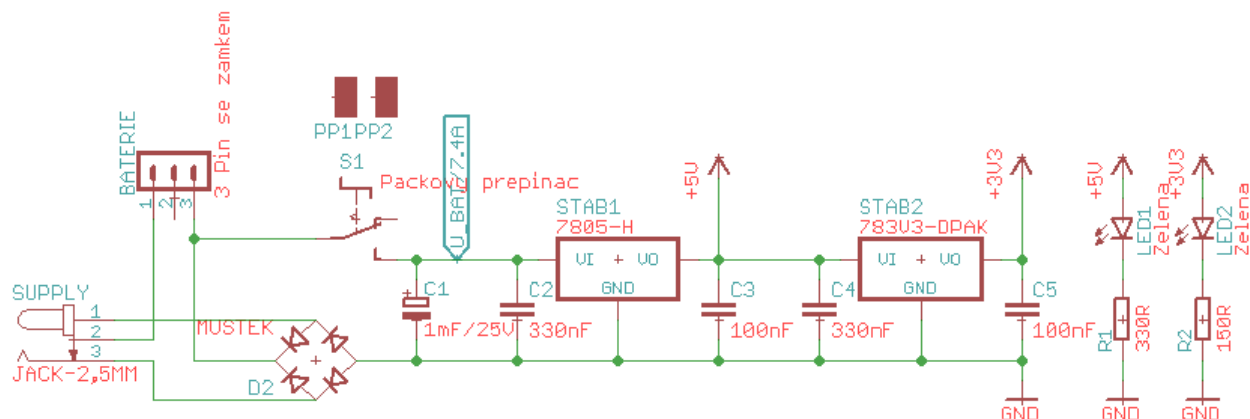
Schéma vývojové desky je rozděleno do osmi pracovních výkresů, které obsahují 323 součástek rozdělených do 30 bloků. Každý blok obsahuje stejné nebo podobné součástky, aby se dalo ve schématu lépe orientovat.

6.3.1 Zdroj

Modul NanoSocketLAN a Bluetooth modul potřebují ke své činnosti 3,3V a ostatní periferie 5V. Toto napětí dodávají dva lineární stabilizátory STAB1 a STAB2. LED diody LED1 a LED2 signalizují přítomné napětí 3,3V a 5V.

Napájení zajišťuje konektor SUPPLY, který slouží pro napájení ze síťového adaptéru 7-12V stejnosměrného i střídavého napětí. Usměrnění zajišťuje diodový můstek D2.

Deska lze napájet také z baterie pomocí konektoru BATERIE. Přepínání mezi baterií a síťovým adaptérem zajišťuje rozpinací kontakt v konektoru SUPPLY. Zapnutí a vypnutí desky se provádí páčkovým přepínačem S1. Schéma zdroje ukazuje obrázek č.33.



Obr. 33 - Schéma zdroje

6.3.2 Přepínač signálu

Vývojová deska obsahuje dvě tlačítka s aretací, pomocí kterých se přepíná UART signál mezi jednotlivými periferiemi. Přepínače jsou umístěny v horní části DPS. Ve schématu jsou označeny jako S24 a S25. K přepínačům je připojen USB/UART převodník, Ethernet/UART převodník, JTAG programátor, Bluetooth modul, mikrokontroléry a piny pro připojení externích zařízení.

Signál se přepíná dle tabulky č.5.

Spínač	Poloha	uP-USB	uP-BT	BT-Piny	JTAG-USB	JTAG-Ethernet	Ethernet-Free	USB-Piny	uP-JTAG
S24	Nahoře	x	ok	-	ok	x	ok	x	x
S25	Nahoře	x	ok	-	x	ok	x	ok	x
S24	Dole	x	x	ok	-	x	ok	x	ok
S25	Dole	ok	x	ok	x	ok	x	-	x

Tab. 5- Tabulka přepínání signálu UART

Spínače S24 a S25 se nastaví do polohy nahoře nebo dole podle toho, jaké periferie se budou využívat. Význam jednotlivých znaků:

- x periferie nelze použít,
- ok periferie je připojena a lze ji využít,
- - Periferie je připojena, ale lze ji využít, pouze pokud se nevyužívá komunikace s MCU.

6.3.3 Mikrokontroléry ATmega

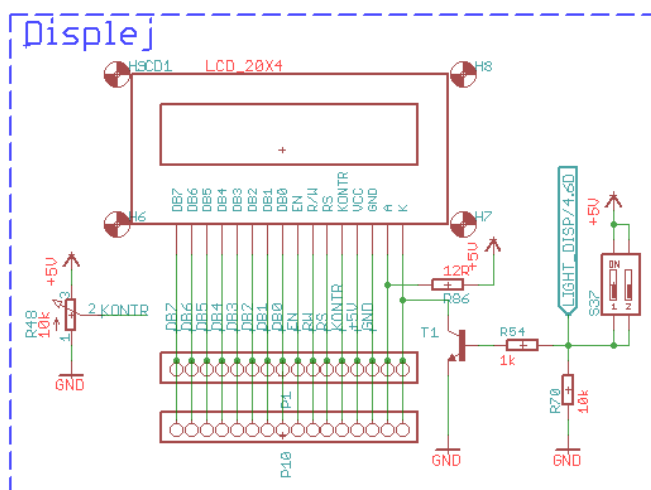
Mikrokontroléry ATmega mají připojeny veškeré vstupně výstupní piny k precizním lištám a poté k dip vypínačům pro připojení externích periferií. Takt mikrokontroléru je dán externím krystalem. K MCU jsou připojeny také podpůrné pasivní prvky dle doporučení výrobce.

6.3.4 LCD displej

Na desce je osazen LCD displej 4*20 znaků s ekvivalentním řadičem k HD44780 od firmy Hitachi. Paměťový prostor má dvě části DDRAM a CGRAM. DDRAM je paměť obsahující znaky aktuálně zobrazené na displeji a CGRAM obsahuje znakovou sadu, tu lze rozšířit o 8 znaků dle potřeby.

Komunikace s displejem může být 8 - bitová nebo 4 - bitová. K MCU je LCD připojen 4 - bitově, lze jej však připojit i 8 - bitově pomocí precizní lišty a vodičů.

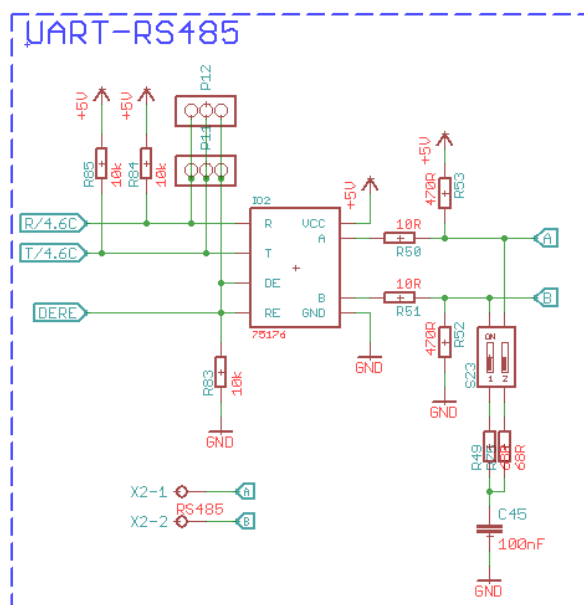
Kontrast LCD displeje se mění trimrem R48. Podsvícení LCD se zapíná vypínačem S37 nebo log. jedničkou na vstup tranzistoru T1. Schéma LCD displeje ukazuje obrázek č.34.



Obr. 34 - Schéma zapojení LCD displeje

6.3.5 Bezzákmitové přepínače a tlačítka

Při stisku tlačítka nebo přepnutí polohy přepínače dojde vždy k překmitu logické úrovně. Aby nedocházelo k zákmitům, vloží se mezi MCU a přepínač RC obvod. Takto jsou na desce ošetřena všechna tlačítka i vypínače. Schéma bezzákmitových spínačů ukazuje obrázek č.35.

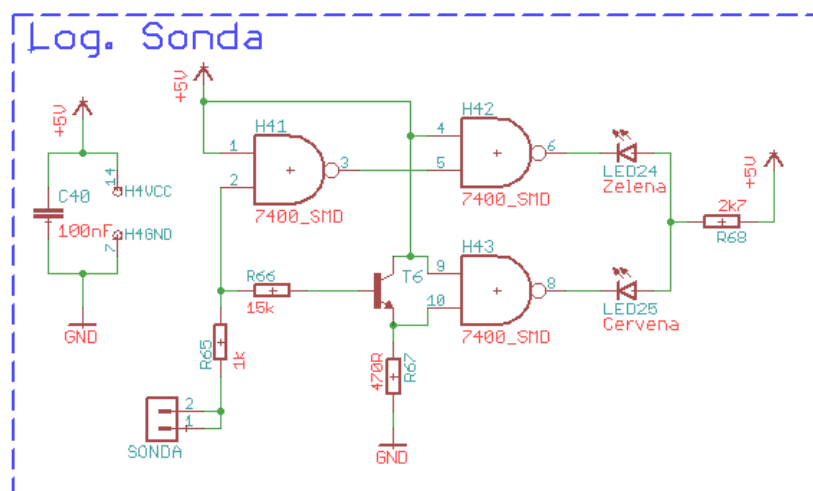


Obr. 36 - Schéma zapojení převodníku UART – RS 485

6.3.7 Logická sonda

Logická sonda je zařízení, jímž lze kontrolovat logické úrovně v testovaném obvodu. Pro obvody postavené na bázi logiky TTL platí, že signál má úroveň log. 1 (neboli H), je-li jeho napětí větší než 2 V, signál má úroveň log. 0 (neboli L), je-li jeho napětí menší než 0,8 V.[19]

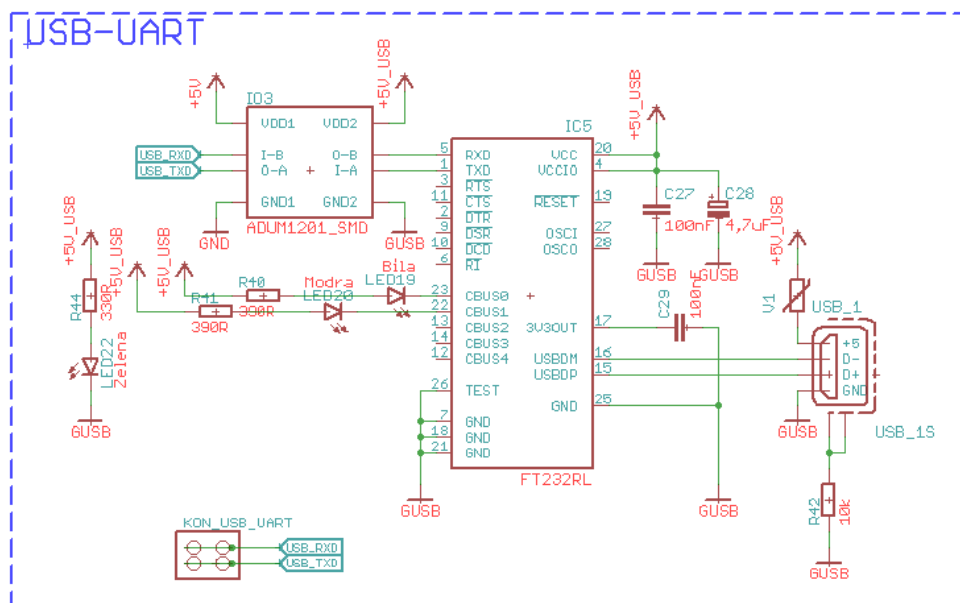
Logická sonda obsahuje dvě LED diody. Zelenou LED24, která signalizuje log. 0 a červenou LED25, která signalizuje log. 1. Nesvítí-li ani jedna z LED diod, není na vstup sondy přiveden žádný signál nebo je napěťová úroveň signálu v zakázaném pásmu 0,8V – 2 V. Svítí-li obě diody, je na vstup sondy přiveden signál s vysokou frekvencí a lidské oko blikání LED diod považuje za stálý svit. Schéma logické sondy ukazuje obrázek č.37.



Obr. 37 - Schéma zapojení logické sondy

6.3.8 USB-UART

Vývojová deska je galvanicky oddělena od PC pomocí dvojitého digitálního izolátoru IO3. Převod signálu z USB na signál UART zajišťuje převodník IC5. K převodníku jsou připojeny dvě LED diody. Modrá LED dioda signalizuje příjem z USB a bílá LED dioda odesílání. Zelená LED dioda signalizuje přítomné napětí 5V z USB portu. Schéma převodníku USB/UART ukazuje obrázek č.38.



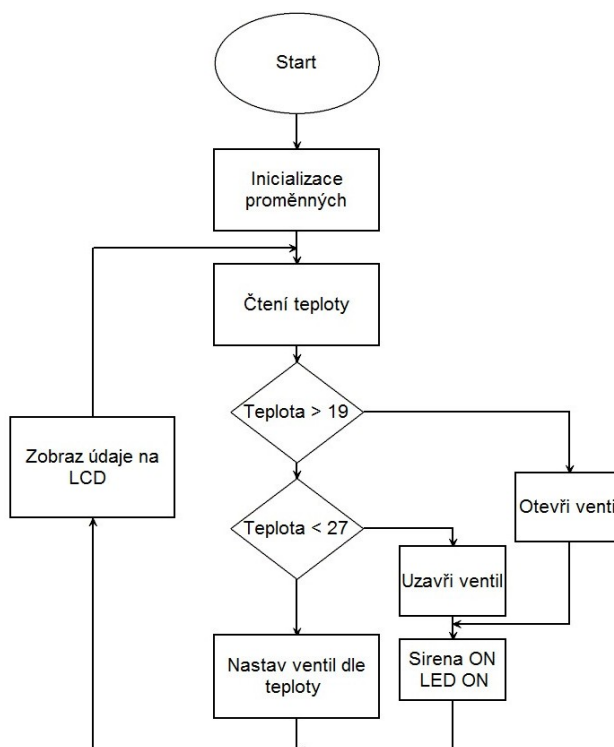
Obr. 38 - Schéma zapojení převodníku USB-UART

7 Test řešení v rámci regulační úlohy

K otestování funkčnosti vzdálené aktualizace firmware i samotné desky je navržena jednoduchá simulace regulace teploty za pomoci digitálního teplotního čidla a servomotoru, který ovládá ventil přivádějící vodu do systému. Vývojový diagram regulační úlohy je na obrázku č.39.

Vestavěný řídicí systém reguluje teplotu na hodnotu 23°C. Jakmile se teplota zvýší, servomotor otevře více ventil, aby potrubím teklo větší množství vody k ochlazení systému. Pokud klesne teplota pod 23°C, ventil se začne přivírat a chladicí médium bude protékat menší rychlostí a systém méně ochlazovat.

V případě překročení teploty nad 27°C bude již ventil úplně otevřen. Rozezní se siréna a rozsvítí červená LED dioda varující obsluhu, že je systém v poruše. Pokud teplota klesne pod 19°C, ventil již bude úplně uzavřen a opět se rozsvítí červená LED dioda a rozezní varovný signál sirény. Aktuální teplota a natočení ventilu se zobrazují na LCD displeji.



Obr. 39 - Vývojový diagram regulace teploty



Obr. 40 - Foto LCD displeje s údaji o teplotě a natočení

8 Návrh knihoven a software pro vývojovou desku

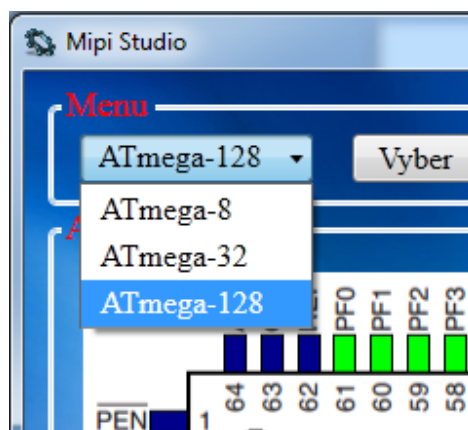
Vývojová deska je zařízení, které slouží HW vývojovému pracovníku jako dokonalý nástroj k testování a zkoušení různého HW a SW ještě dříve než vyrobí finální zařízení.

Jelikož se nemusí zabývat výrobou HW a může ihned začít programovat firmware zařízení, ušetří čas i náklady. To vede k zefektivnění výroby a celého návrhu prototypových zařízení.

8.1 Software pro generování obslužných rutin a nastavení vývojové desky

Práce s vývojovou deskou vede k opakujícím se úkonům a nastavením. Toto nastavení lze sloučit do jednoduchého ovládacího software, který za nás vygeneruje obslužné rutiny, nastavení vstupně výstupních portů, atd..

Veškeré nastavení se provádí v přehledném grafickém prostředí. V horní části se nachází menu, pomocí kterého lze z roletové nabídky (obrázek č.41) vybrat MCU které chceme nastavit. Výběr potvrdíme tlačítkem Vyber. Zobrazí se nám daný MCU a prvky pro jeho nastavení viz obrázek č.42.

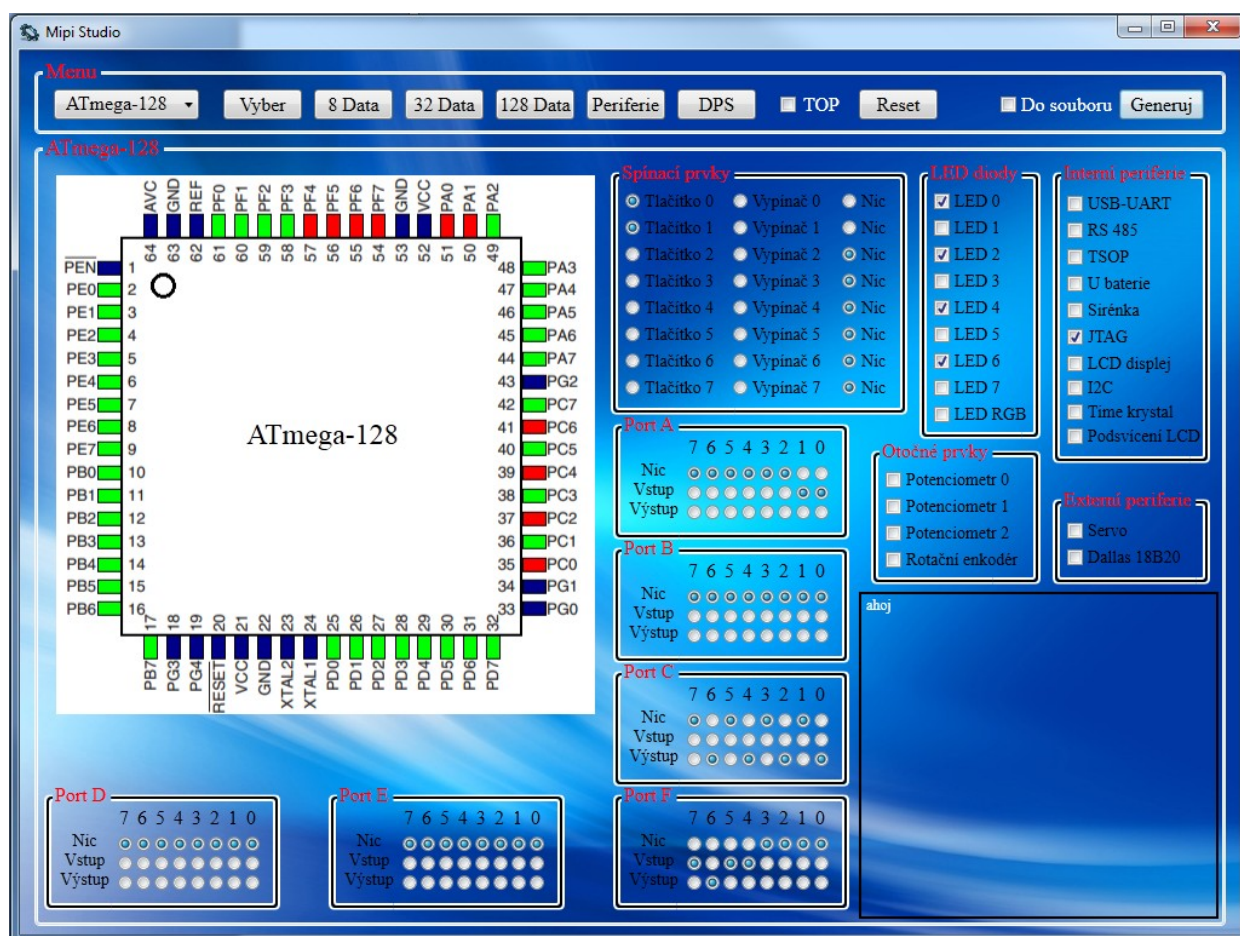


Obr. 41 - Menu pro výběr MCU

Při použití některé z periférií se automaticky zvýrazní červeně I/O pin signalizující jeho využití. Zelenou barvou jsou značeny volné I/O piny. Modrá barva pinů signalizuje připojení předem již daných periférií nebo napájení.

V pravém dolním rohu se nachází textové pole, ze kterého lze zkopírovat vygenerovaný text. Text můžeme generovat do tohoto pole nebo textového souboru.

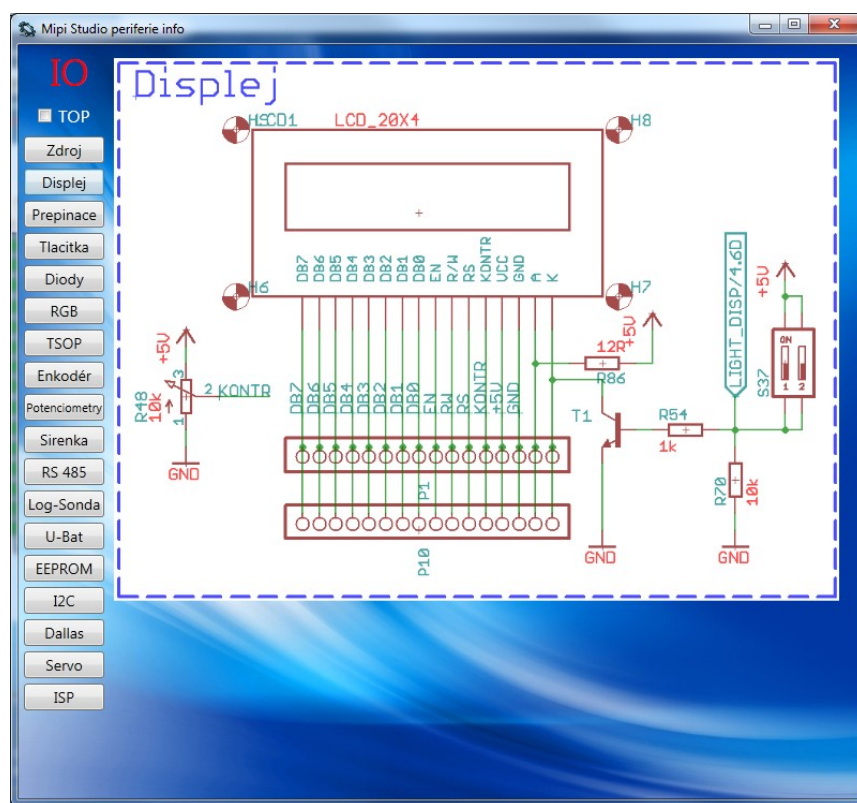
Zatržením některé z položek v seznamu periférií se po stisku tlačítka generuj, vygeneruje nastavení I/O portů jako vstupních (výstupních), obslužná rutina pro danou periférii a komentáře k danému kódu, aby jej bylo snadné modifikovat.



Obr. 42 - Software Mipi Studio

Menu obsahuje tlačítka 8 Data, 32 Data, 128 Data a Periferie. Stisknutím se zobrazí nové okno s nápovědou (obrázek č.43) pro daný MCU nebo periferie obsažené na desce. Checkbox s názvem TOP zajišťuje, aby okno zůstalo v popředí a ostatní okna a programy zůstávaly pod tímto oknem.

Obslužný software je psán v jazyce XAML a C#. XAML (Extensible Application Markup Language) je značkovací jazyk (obdoba HTML) využívaný k popisu grafického rozhraní v aplikacích společnosti Microsoft nové generace. Je vyvíjený Microsoftem a založený na XML. Zkratka původně znamenala Extensible Avalon Markup Language, kde Avalon bylo kódové označení pro Windows Presentation Foundation. C# je vysokoúrovňový objektově orientovaný programovací jazyk vyvinutý firmou Microsoft zároveň s platformou .NET Framework. Microsoft založil C# na jazycích C++ a Java (a je tedy nepřímým potomkem jazyka C, ze kterého čerpá syntaxi). [20] [21]



Obr. 43 - Info o perifériách použitých na vývojové desce

8.2 Web server s nápovědou

Webový server je počítač, mikrokontrolér, VŘS nebo jakýkoliv podobný systém, který je zodpovědný za vyřizování požadavků HTTP od klientů - programů zvaných webové prohlížeče. Vyřízením požadavků se rozumí odeslání webové stránky. Webové stránky jsou obvykle dokumenty HTML. Jednotlivé webové servery se mohou v různých jednotlivostech značně lišit. Přesto mají několik společných vlastností.

Každý webový server je připojen k počítačové síti a přijímá požadavky ve tvaru HTTP. Tyto požadavky vyřizuje a počítač, který požadavek vznesl, vrací odpověď. Odpověď obvykle představuje nějaký HTML dokument. Může to být ale i dokument v jiném formátu - text, obrázek apod. Součástí odpovědi je i tzv. Stavový kód odpovědi. Ten udává, zda byl požadavek vyřízen v pořádku, či zda došlo k nějakým obtížím.[22]

Modul NanoSocketLAN umožňuje nahrát do modulu vlastní, uživatelem vytvořený web server s maximální velikostí 256kByte. Nejprve se napíšou webové stránky v jazyce HTML. Následuje spuštění programu iChipConfig a vygenerování souboru *.img. Tento soubor nahrajeme do modulu a povolíme v menu web server a nastavíme mu IP adresu. Vývojová deska má nastavenou IP adresu 192.168.0.150, pokud tuto adresu zadáme do internetového prohlížeče, zobrazí se nám webové stránky uložené v modulu NanoSocketLAN které obsahují nápovědu k vývojové desce.

9 Závěr

Tato diplomová práce se zabývala problematikou vzdálené aktualizace firmware mikrokontrolérů AVR. Dále pak návrhem a realizací vývojové desky s ethernetovou konektivitou tak, aby byla schopna nejen vzdáleně aktualizovat firmware, ale také ji otestovat na jednoduché regulační úloze.

Počáteční práce se zabývala architekturou mikrokontrolérů AVR a porovnáním různých variant programování těchto mikrokontrolérů. Z úvodní studie vyplynulo, že nejpoužívanější metodou programování mikrokontrolérů AVR je sériové programování ISP. Avšak nejvhodnější variantou k vzdálené aktualizaci je JTAG programování díky možnosti nejen mikrokontrolér naprogramovat, ale také jej ladit, zapisovat a vyčítat hodnoty z registrů.

Připojení k JTAG programátoru a zároveň připojení k vzdálenému PC zajišťuje modul NanoSocketLAN. Miniaturní modul NanoSocketLAN, od firmy ConnectOne, je určený ke snadnému připojení jakékoli aplikace do sítě 10/100BaseT Ethernet LAN. Modul dovoluje nasazení LAN konektivity, bez složitého programování TCP/IP stacku, nebo potřeby ovladače pro mikrokontrolér. K mikrokontroléru AVR se připojujeme pomocí rozhraní UART.

Následovala analýza potřebného software a jeho nastavení. Poté byl vytvořen JTAG programátor a otestována vzdálená aktualizace firmware. Test zařízení proběhl úspěšně a je popsán v kapitole Praktická ukázka práce s vytvořeným hardware.

Po nastudování problematiky vzdálené aktualizace proběhl návrh vývojové desky v programu Eagle. V příloze se nachází kompletní elektrické schéma, DPS, osazovací plány, potisk i závěrečné foto vývojové desky.

Nakonec byl napsán software pro generování obslužných rutin desky v jazyce XAML a C#. Pomocí něj byl napsán firmware regulační úlohy a vzdáleně přes ethernet umístěn do paměti mikrokontroléru.

Možnosti zdokonalení systému do budoucna může být připsání nastavování vnitřních periférií mikrokontroléru do nastavovacího software, popřípadě vytvoření přídavných modulů, které se dají do desky připojit pomocí předpřipravených konektorů.

10 Použitá literatura

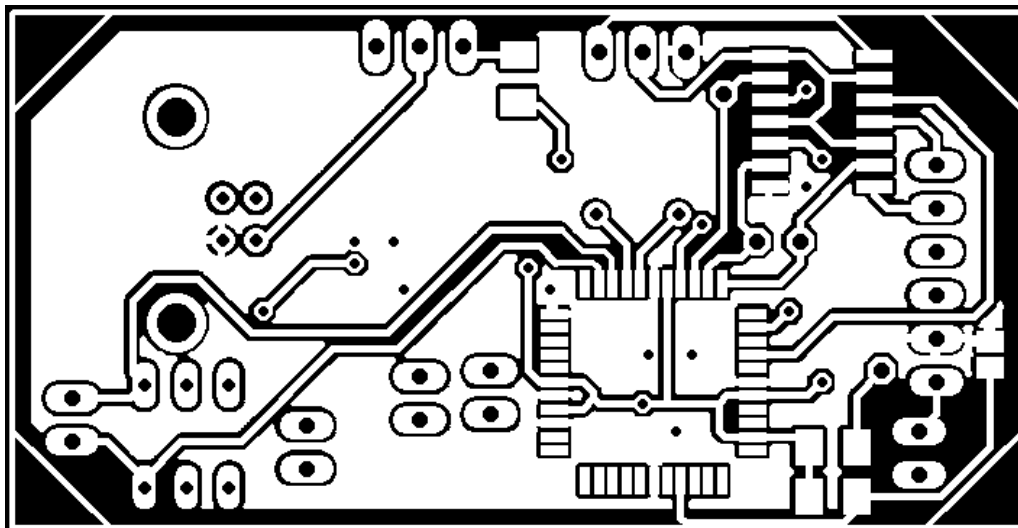
- [1] AVR. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2013-05-04]. Dostupné z: <http://cs.wikipedia.org/wiki/AVR>
- [2] Harvardská architektura. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2013-04-28]. Dostupné z: http://cs.wikipedia.org/wiki/Harvardsk%C3%A1_architektura
- [3] RISC. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2013-04-28]. Dostupné z: cs.wikipedia.org/wiki/RISC
- [4] Firmware. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2013-04-28]. Dostupné z: cs.wikipedia.org/wiki/Firmware
- [5] VÁŇA, Vladimír. *Mikrokontroléry ATMELE AVR: popis procesorů a instrukční soubor*. 1. vyd. Praha: BEN - technická literatura, 2003, 335 s. ISBN 80-730-0083-0
- [6] Obnovení konfiguračních bitů obvodů AVR. *Pandatron.cz - Elektrotechnický magazín* [online]. 4.6.2010 [cit. 2013-04-28]. Dostupné z: http://pandatron.cz/?1452&obnoveni_konfiguracnich_bitu_obvodu_avr
- [7] Popis rozhraní JTAG. *Popis rozhraní JTAG* [online]. 2004 [cit. 2013-04-28]. Dostupné z: <http://noel.feld.cvut.cz/vyu/ap2/JTAGmoje.htm>
- [8] Joint Test Action Group. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2013-04-28]. Dostupné z: cs.wikipedia.org/wiki/JTAG
- [9] STK500 Protocol AVR Bootloader. In: *HW.cz | Vše o elektronice a programování* [online]. 23. Zář 2003 [cit. 2013-04-28]. Dostupné z: <http://www.hw.cz/navrh-obvodu/software/stk500-protocol-avr-bootloader.html>
- [10] Bootloader (zavaděč). In: *Bootloader (zavaděč) - JS Homepage* [online]. [cit. 2013-04-28]. Dostupné z: <http://www.smrz.chrudim.cz/bootloader/>
- [11] KOTZIAN, Jiří. *Navrhování a realizace vestavěných řídicích systémů*. Ostrava, 2012, 179 s.
- [12] *Stroke computer gadgets*. 2013. Dostupné z: http://www.iconshock.com/icons/stroke/computer_gadgets/cat5_ethernet_cable-icon.html
- [13] VERNER, Lukáš. *Autonomní programátor mikroprocesorů AVR s ethernetovou konektivitou* [online]. Brno, 2011 [cit. 2013-04-28]. 57 l. Dostupné z: <https://dspace.vutbr.cz/handle/11012/6188>. Diplomová práce. Vysoké učení technické v brně.
- [14] Miniaturní Ethernet moduly Nano LANReach™ a Nano SocketLAN™. In: *Spezial Electronic - elektronické součástky* [online]. [2013] [cit. 2013-04-28]. Dostupné z: <http://www.spezial.cz/connectone/connect-one-nano-lanreach-uart-usb-spi-ethernet-modul.html>

- [15] CONNECTONE. *Nano SocketLAN*. USA, July 2009.
- [16] HW VSP3 - Virtual Serial Port. In: *HW Group* [online]. [2013] [cit. 2013-04-28]. Dostupné z: http://www.hw-group.com/products/hw_vsp/index_cz.html
- [17] Virtuální TCP/IP COM port pro přenos sériové linky z WiFi a LAN modulů, příklad s programem HW VSP3. In: *Spezial Electronic - elektronické součástky* [online]. [2013] [cit. 2013-04-28]. Dostupné z: <http://www.spezial.cz/apps/virtualni-seriovy-tcp-ip-com-port-redirector.html>
- [18] RS-485. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2013-04-28]. Dostupné z: <http://cs.wikipedia.org/wiki/RS-485>
- [19] Logická sonda. In: *Pandatron.cz - Elektrotechnický magazín* [online]. 2013 [cit. 2013-05-05]. Dostupné z: http://pandatron.cz/?126&logicka_sonda
- [20] Extensible Application Markup Language. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2013-04-28]. Dostupné z: https://cs.wikipedia.org/wiki/Extensible_Application_Markup_Language
- [21] C Sharp. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2013-04-28]. Dostupné z: http://cs.wikipedia.org/wiki/C_Sharp
- [22] Webový server. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2013-05-05]. Dostupné z: http://cs.wikipedia.org/wiki/Webov%C3%BD_server
- [23] Vytváření knihovny součástek. In: *EAGLE Help: Vytváření knihovny součástek* [online]. 2001 [cit. 2013-04-28]. Dostupné z: <http://www.eagle.cz/help/10.htm>

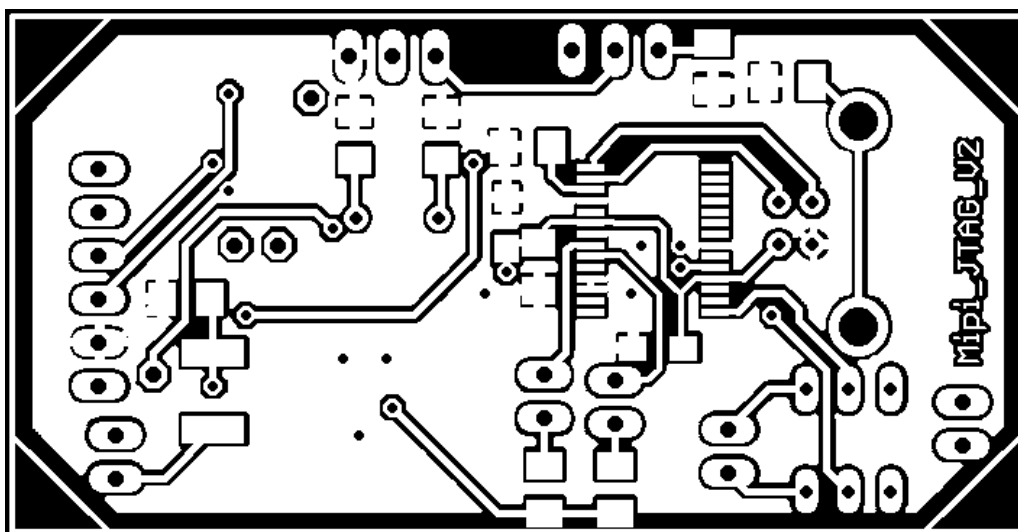
11 Seznam příloh

I.	DPS JTAG programátoru	I
II.	Osazovací plán JTAG programátoru	II
III.	Elektronické schéma vývojové desky	III
IV.	DPS vývojové desky	XI
V.	Osazovací plán vývojové desky	XIII
VI.	Potisk vývojové desky	XV
VII.	Foto vývojové desky	XVI
VIII.	Elektrické schéma redukce bluetooth modulu	XVIII
IX.	DPS redukce bluetooth modulu	XIX
X.	Osazovací plán redukce bluetooth modulu	XX
XI.	Instalace nezbytného software	XXI
XII.	Knihovny pro program Eagle	XXVI
XIII.	Menu pro program Eagle	XXVII

I. DPS JTAG programátoru

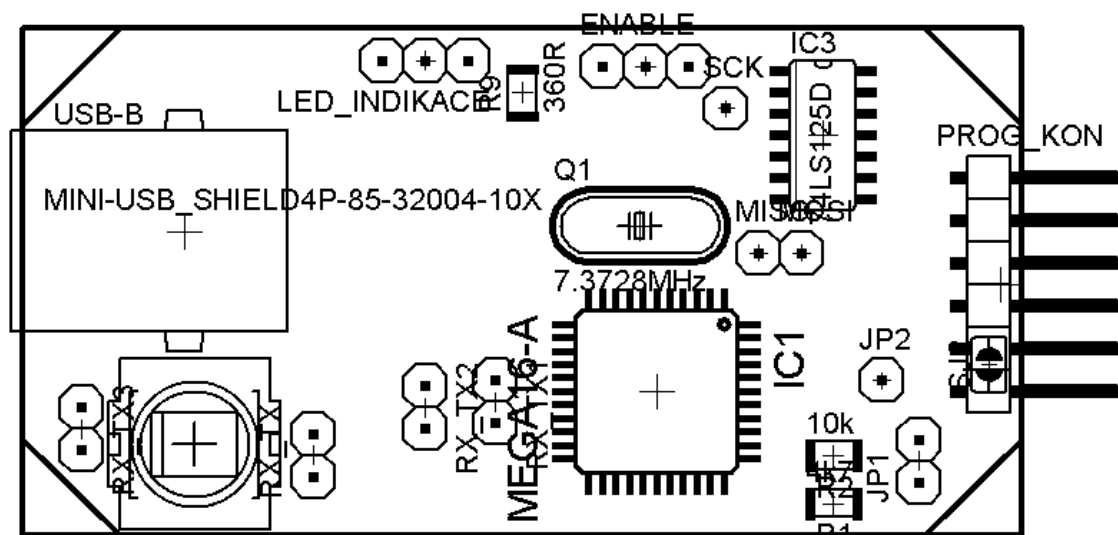


Obr. 44 - DPS JTAG programátoru strana TOP

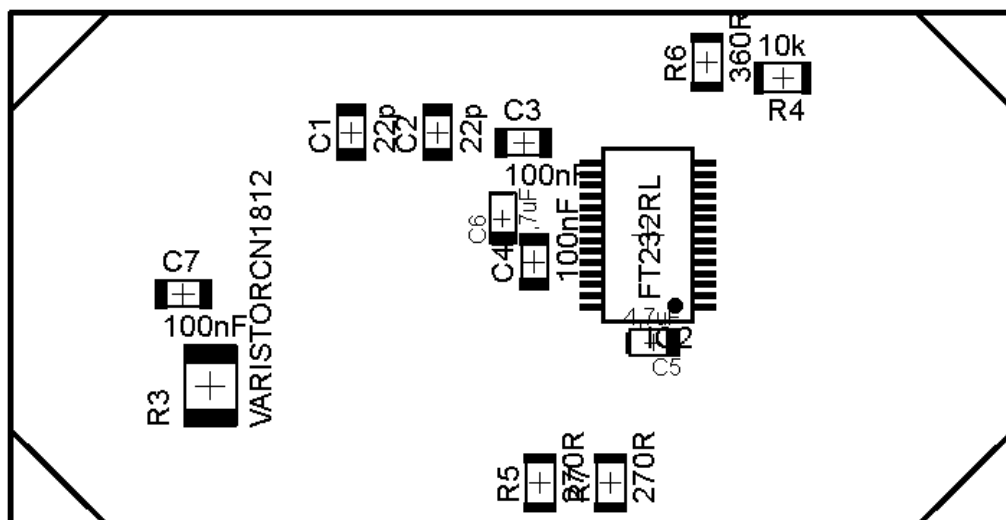


Obr. 45 - DPS JTAG programátoru strana BOTTOM

II. Osazovací plán JTAG programátoru

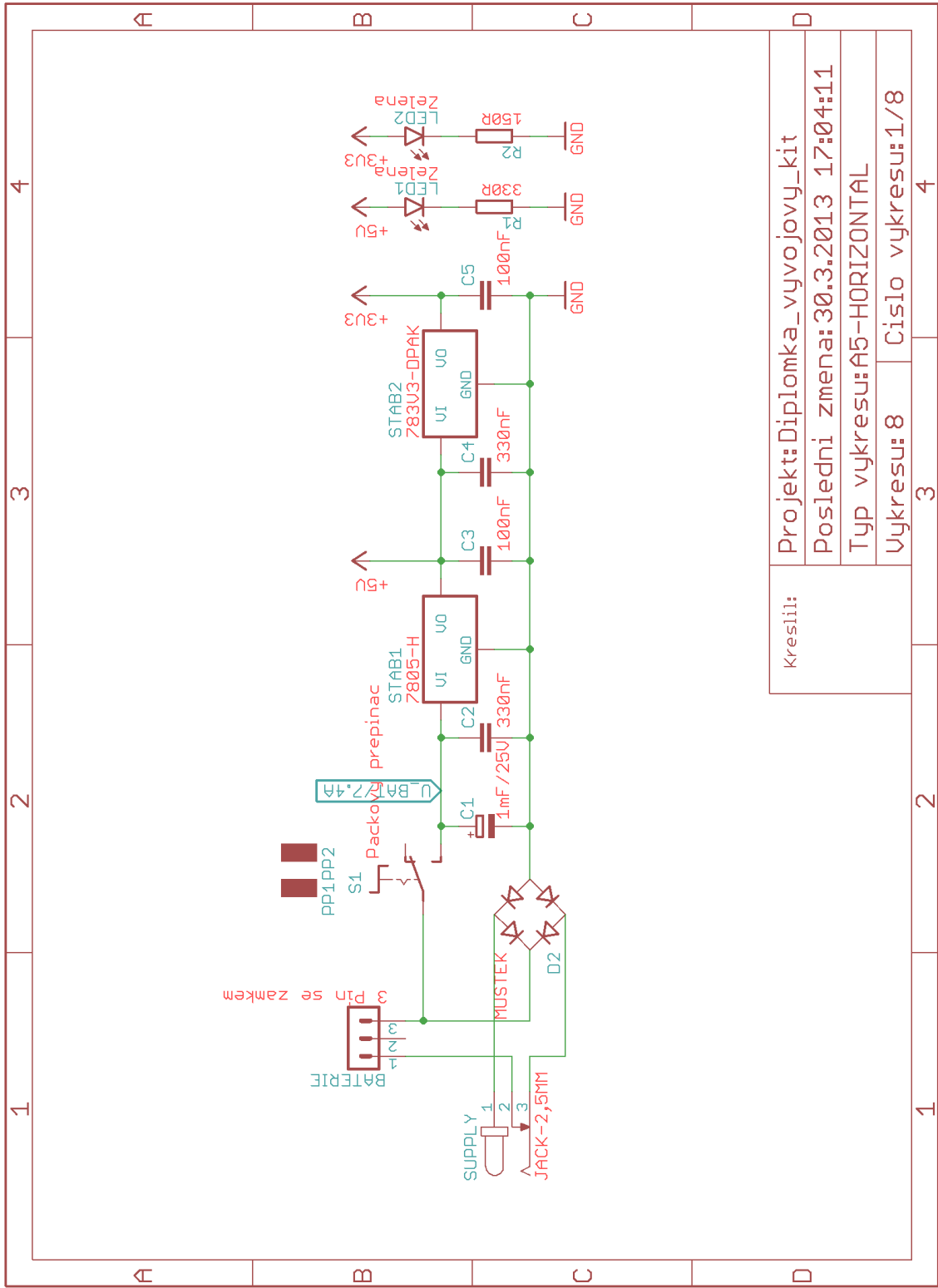


Obr. 46 - Osazovací plán JTAG programátoru strana TOP

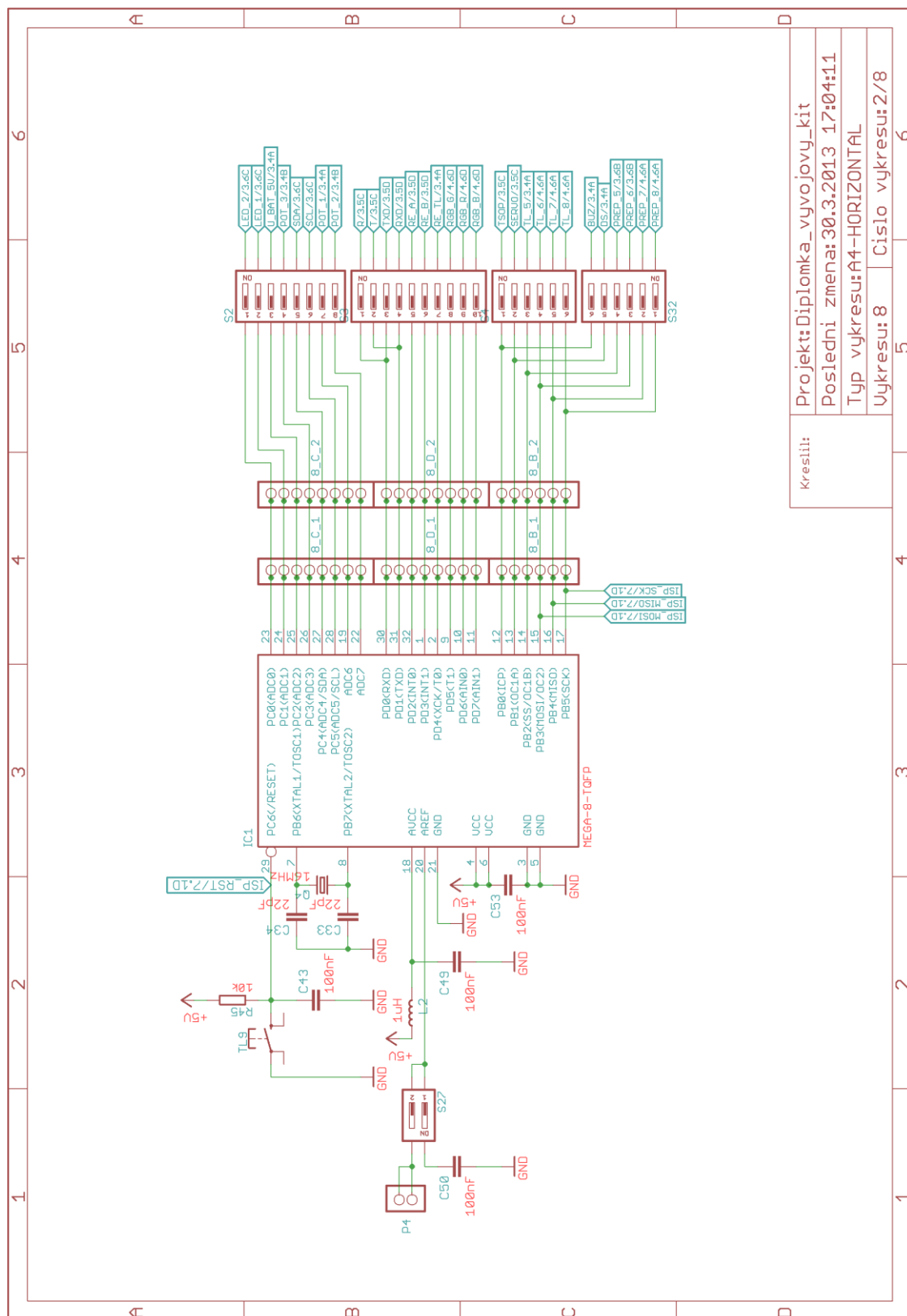


Obr. 47 - Osazovací plán JTAG programátoru strana BOTTOM

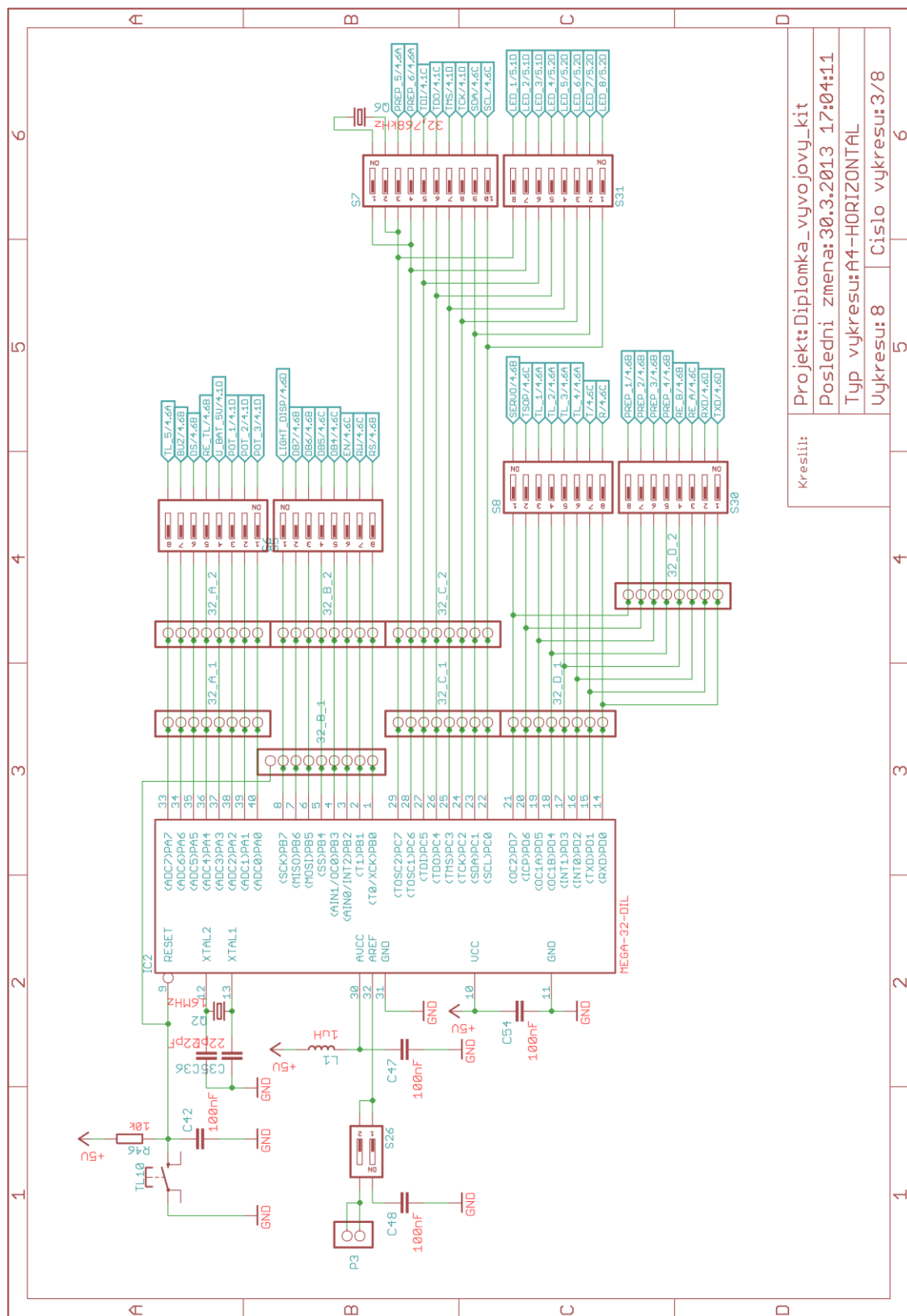
III. Elektronické schéma vývojové desky



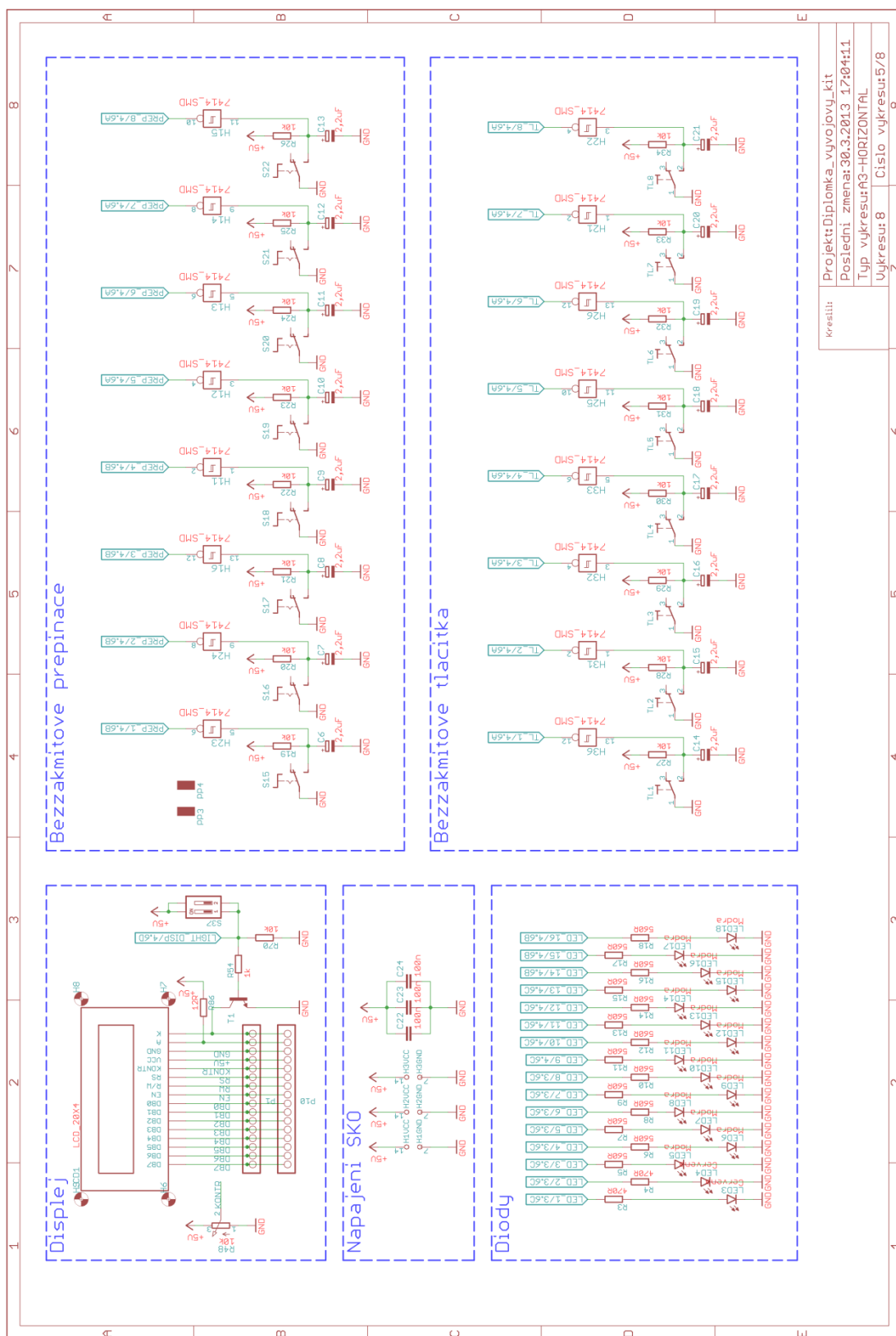
Obr. 48 - Schéma vývojové desky výkres č. 1



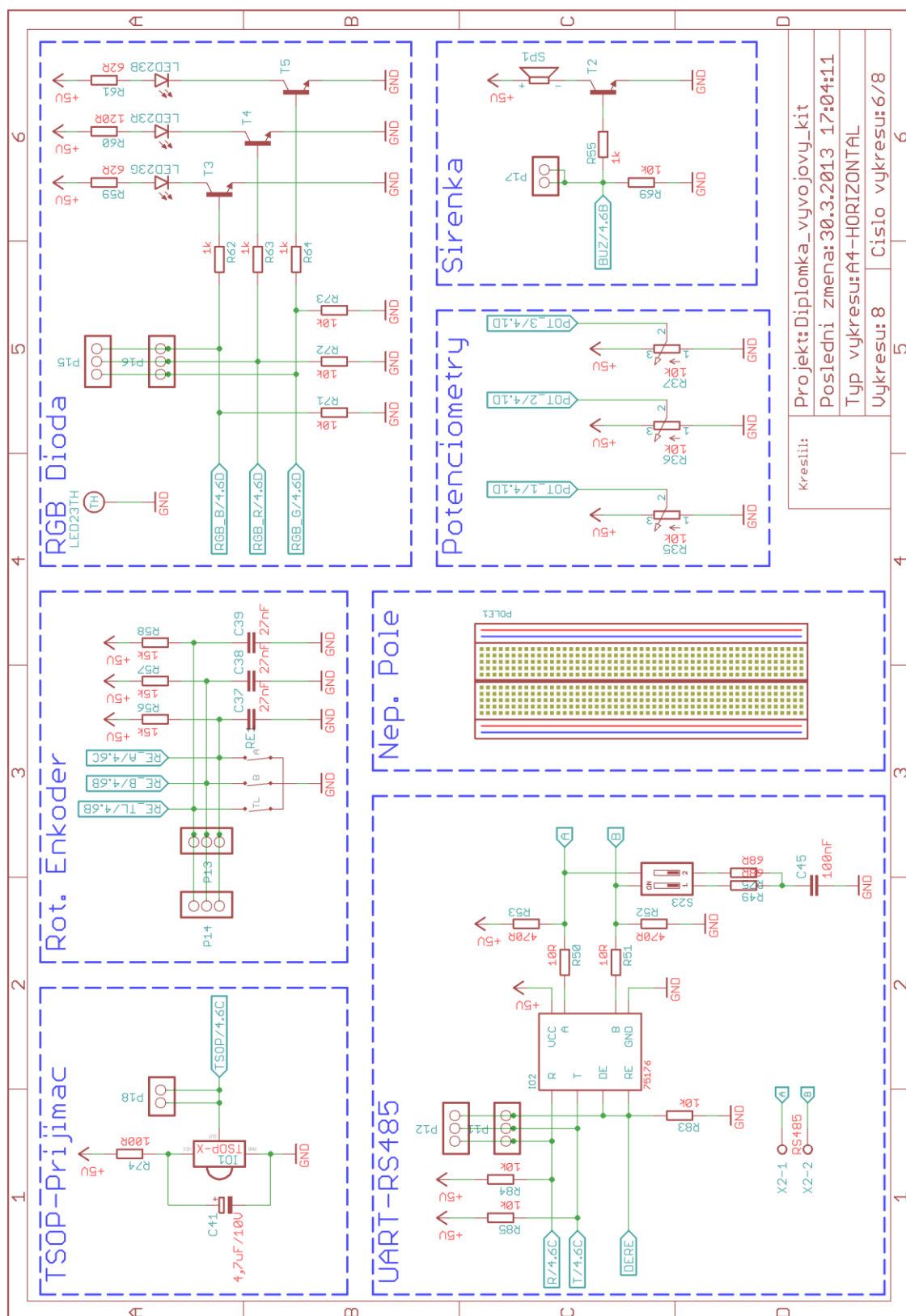
Obr. 49 - Schéma vývojové desky výkres č. 2



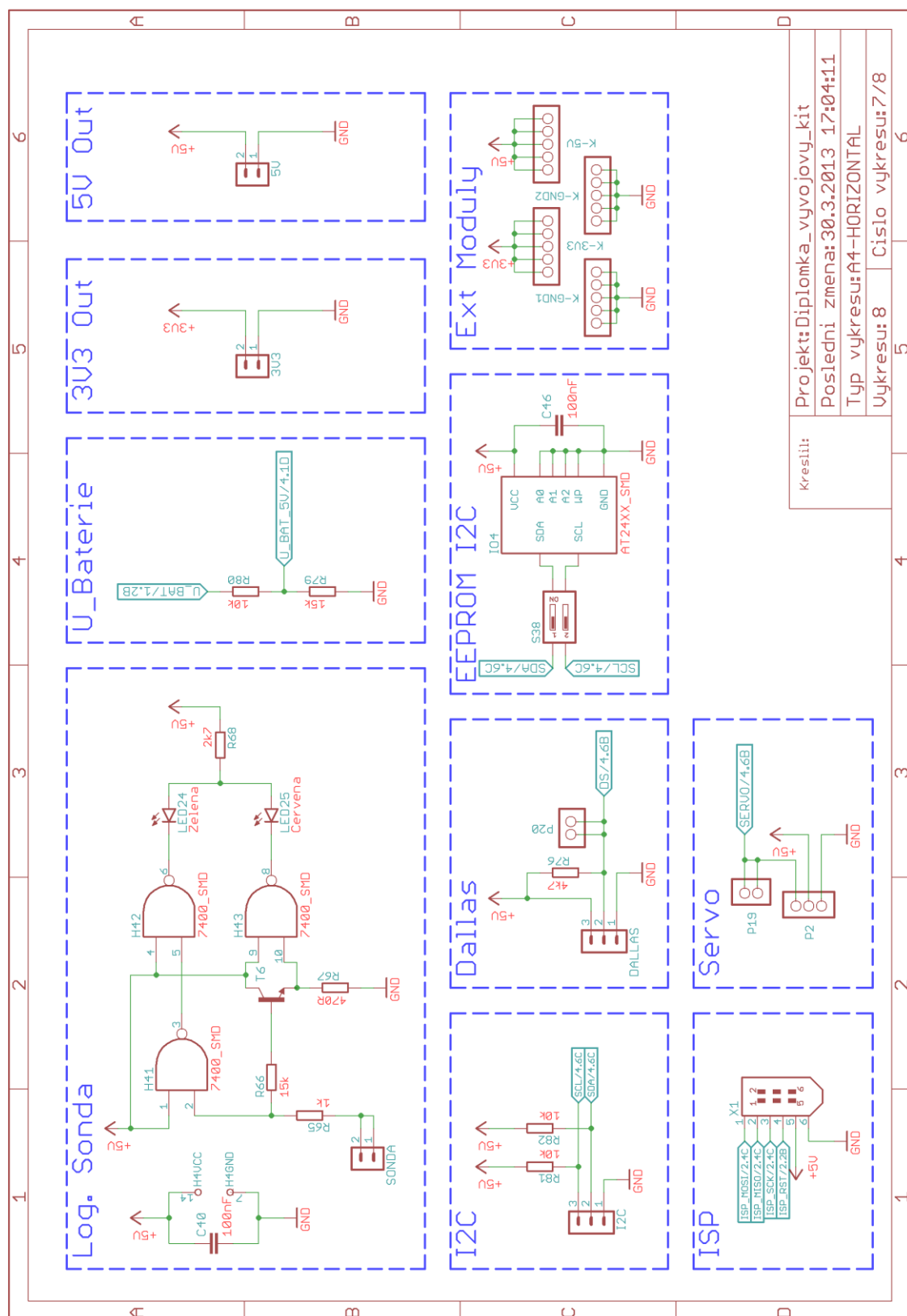
Obr. 50 - Schéma vývojové desky výkres č. 3



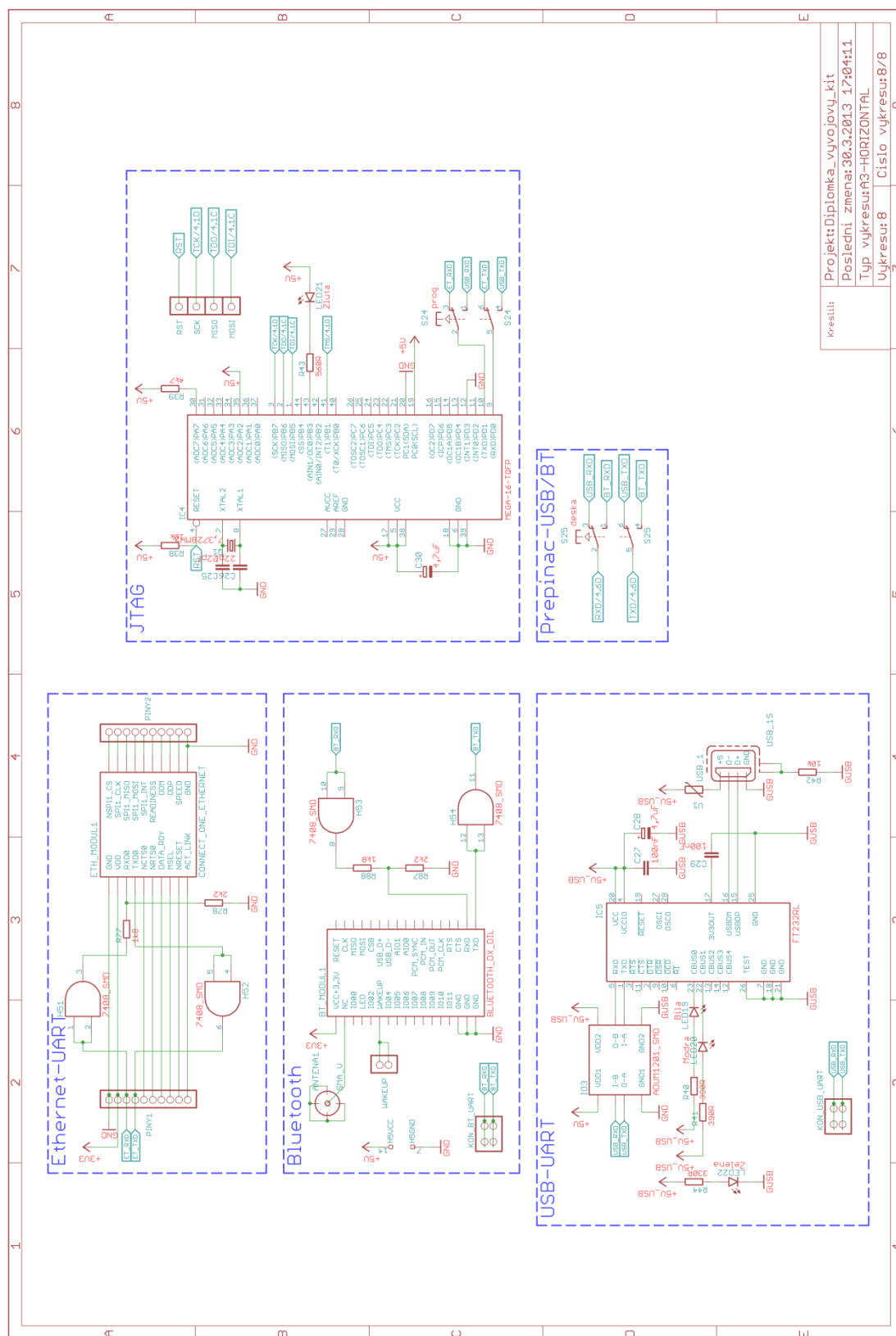
Obr. 52 - Schéma vývojové desky výkres č. 5



Obr. 53 - Schéma vývojové desky výkres č. 6

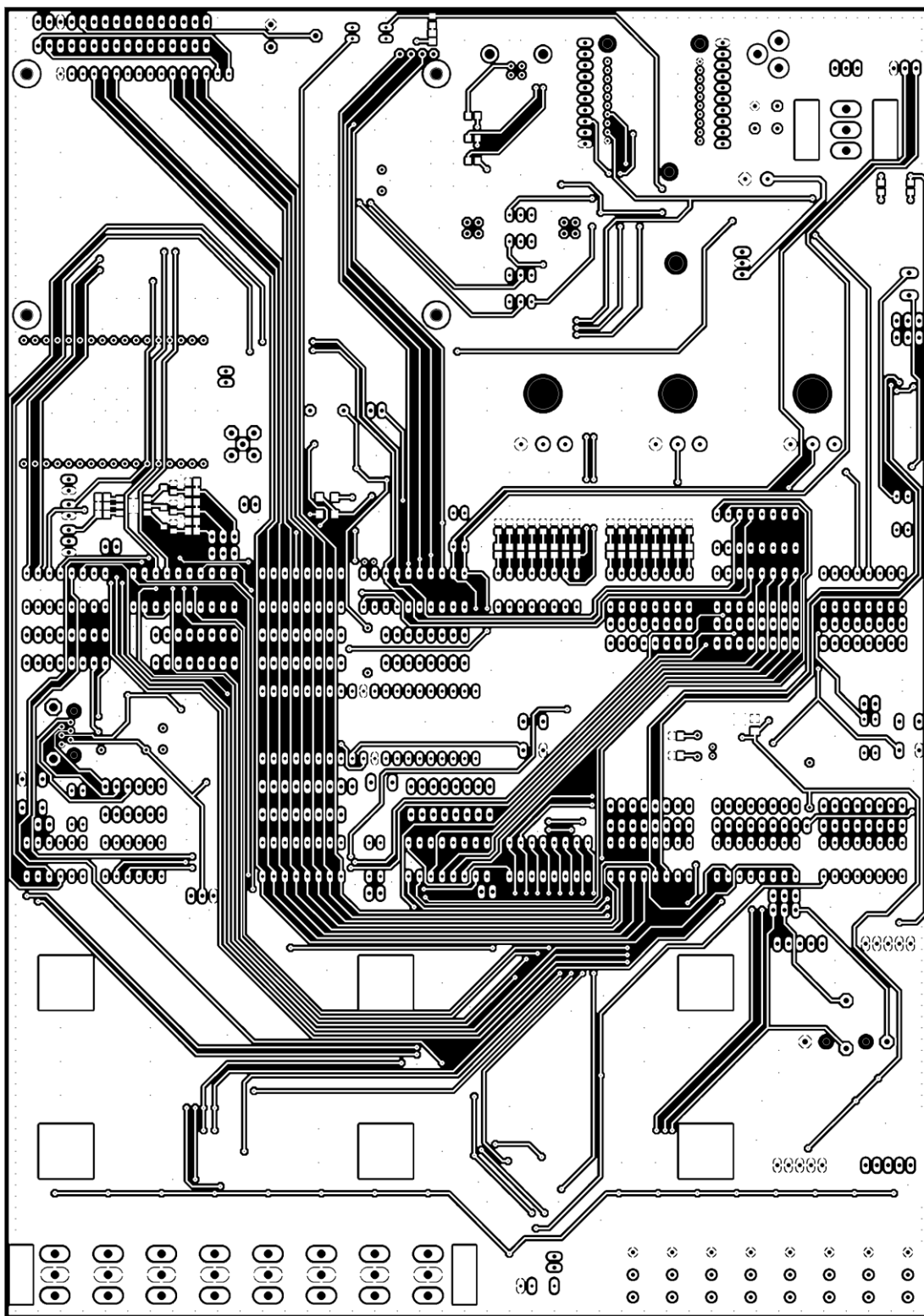


Obr. 54 - Schéma vývojové desky výkres č. 7

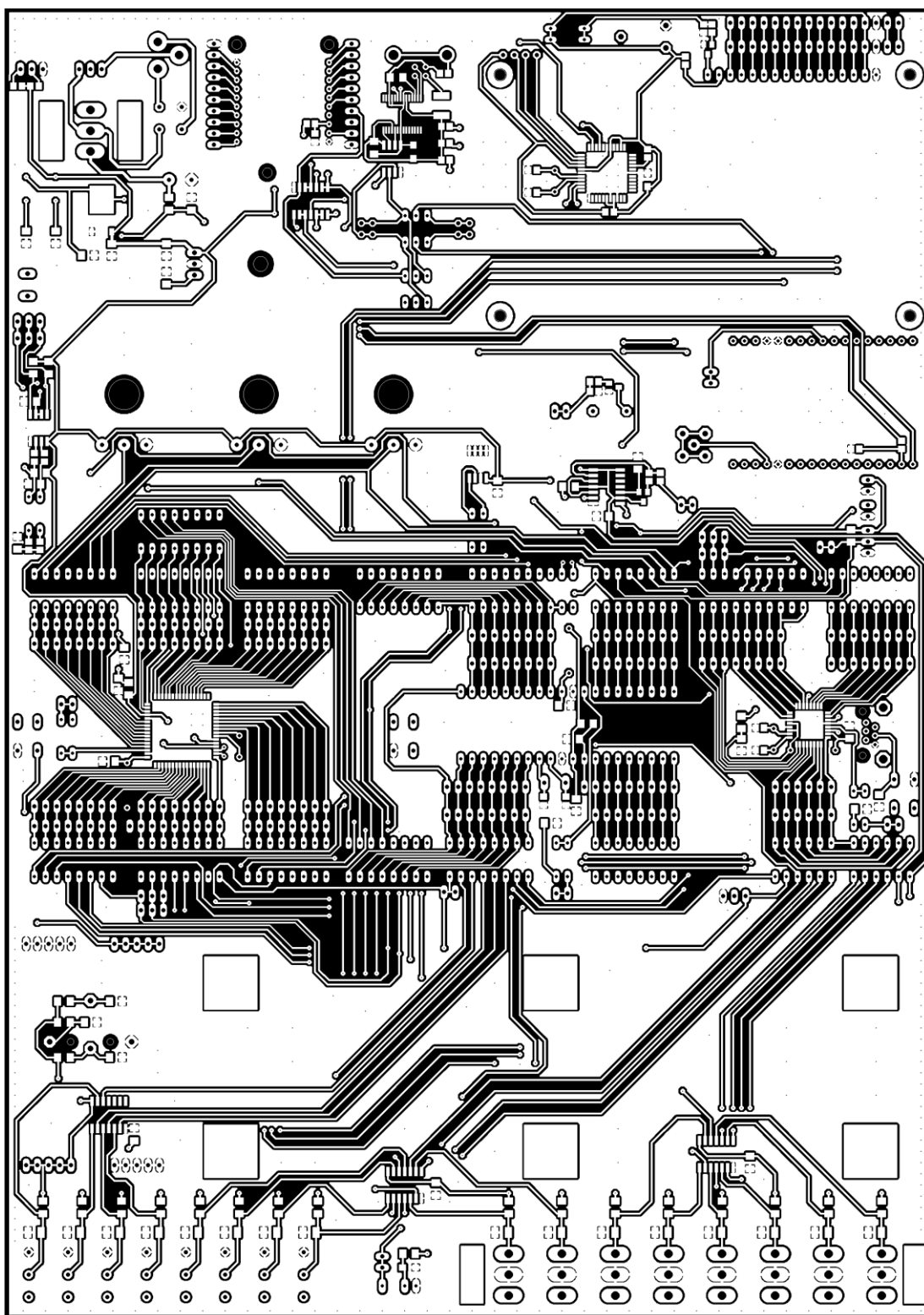


Obr. 55 - Schéma vývojové desky výkres č. 8

IV. DPS vývojové desky

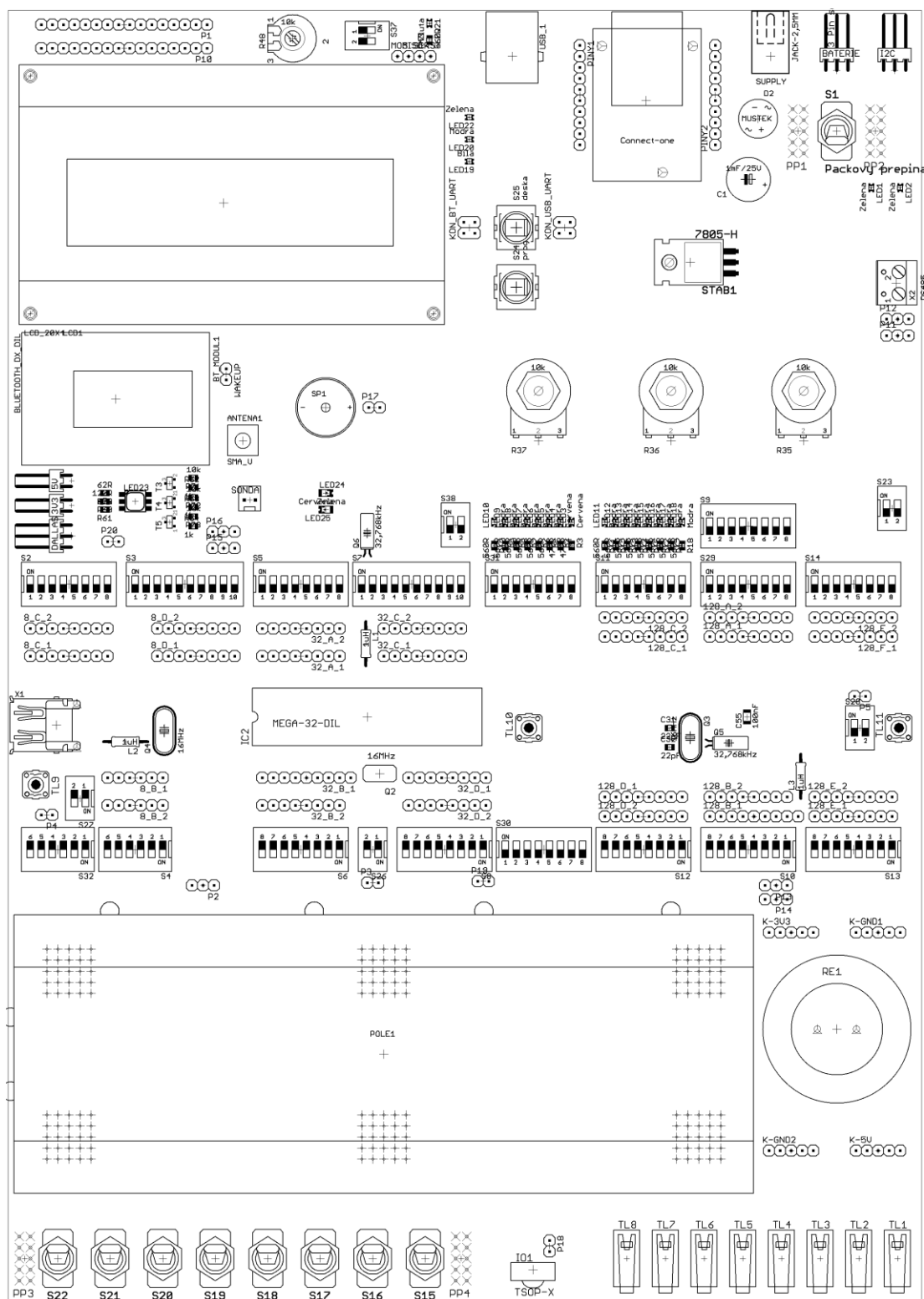


Obr. 56 - DPS vývojové desky strana TOP

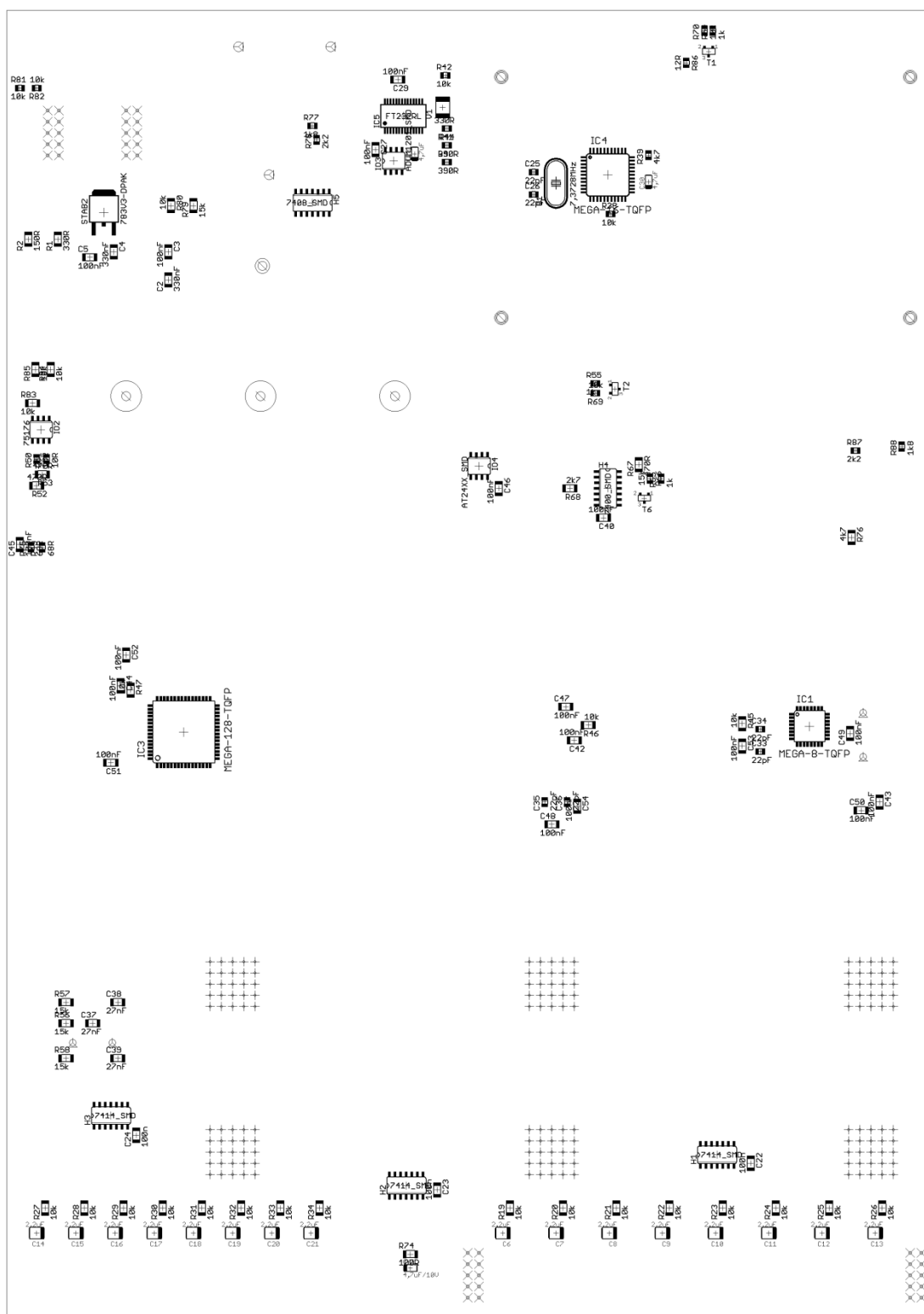


Obr. 57 - DPS vývojové desky strana BOTTOM

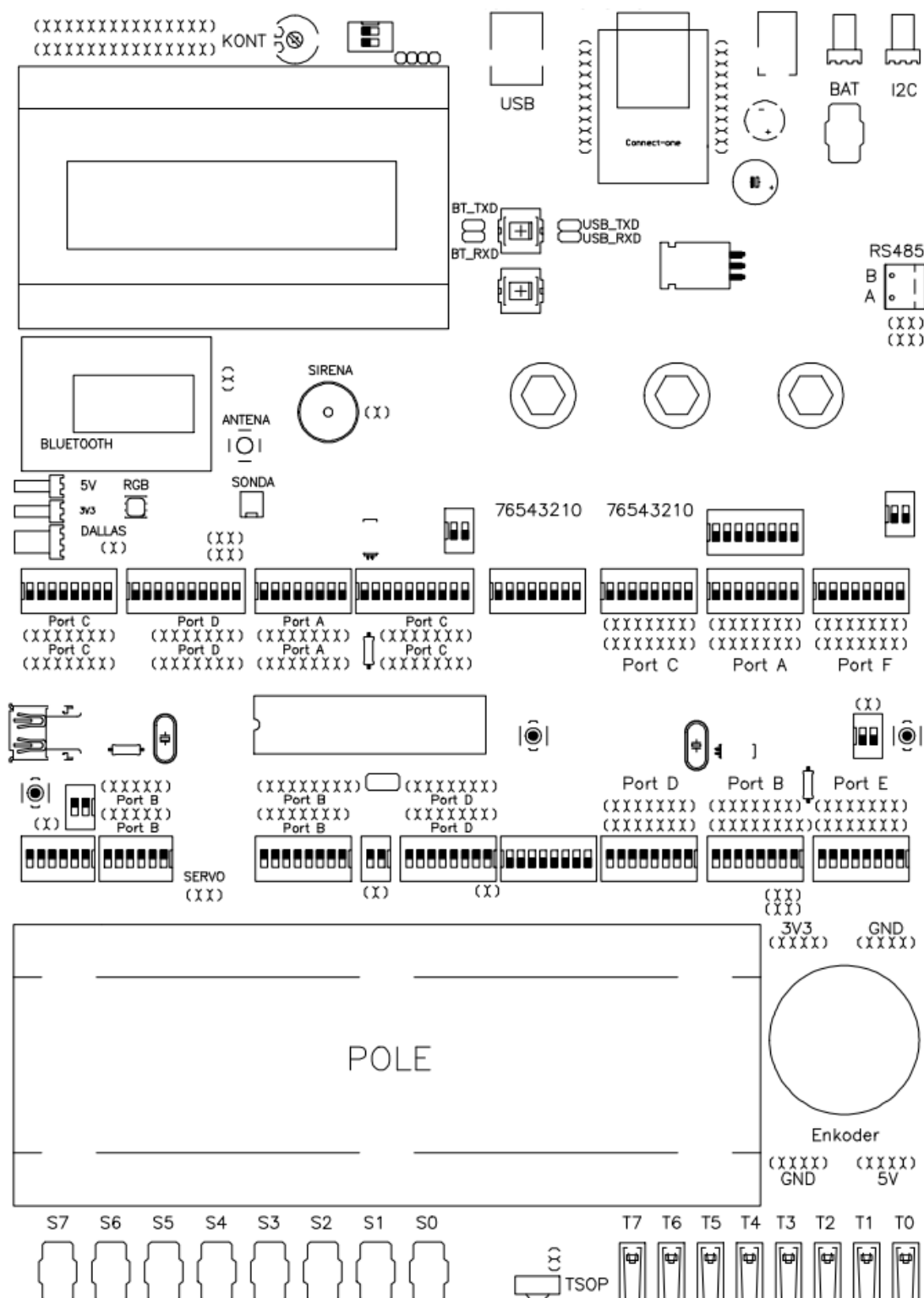
V. Osazovací plán vývojové desky



Obr. 58 - Osazovací plán vývojové desky strana TOP

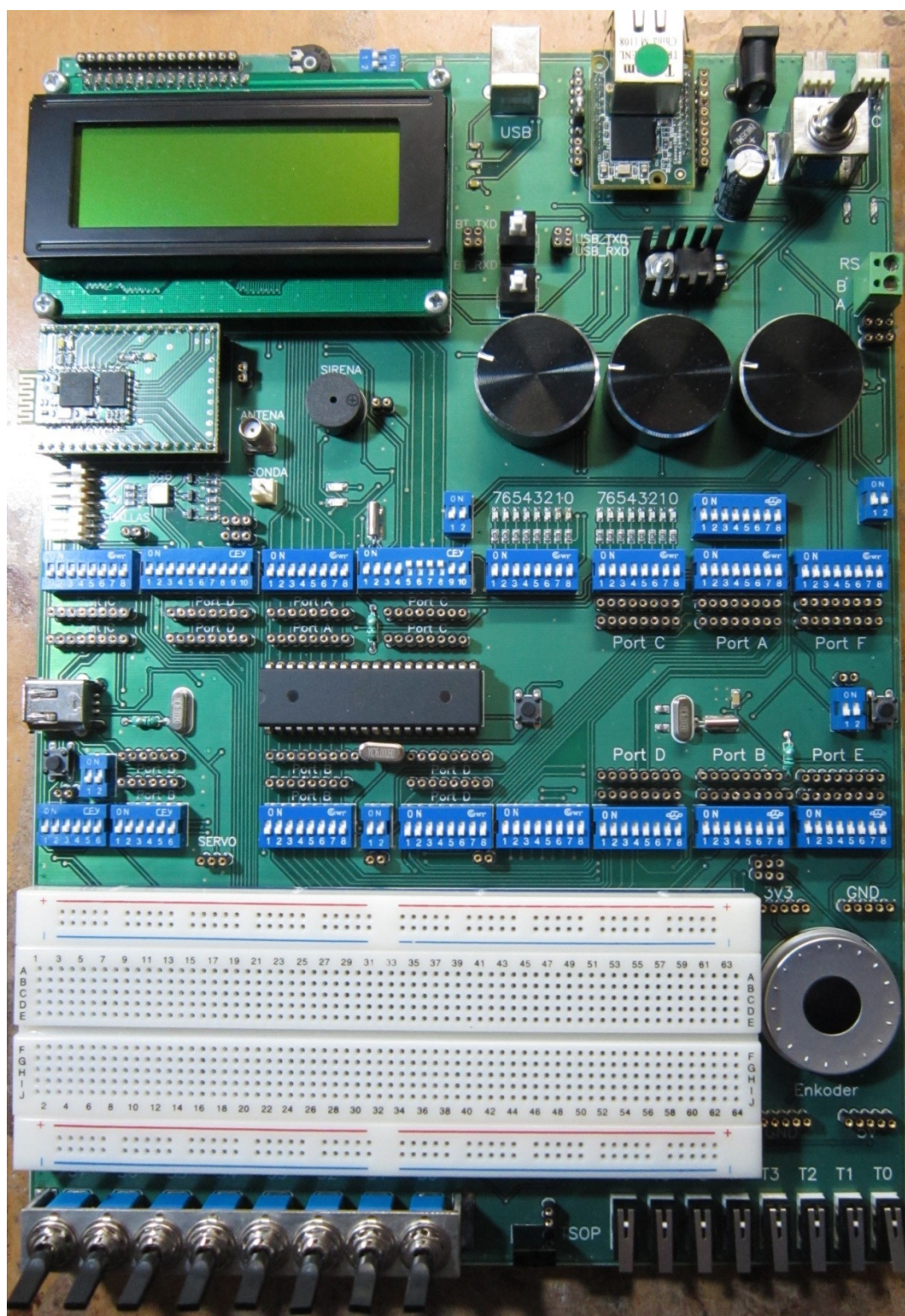


VI. Potisk vývojové desky

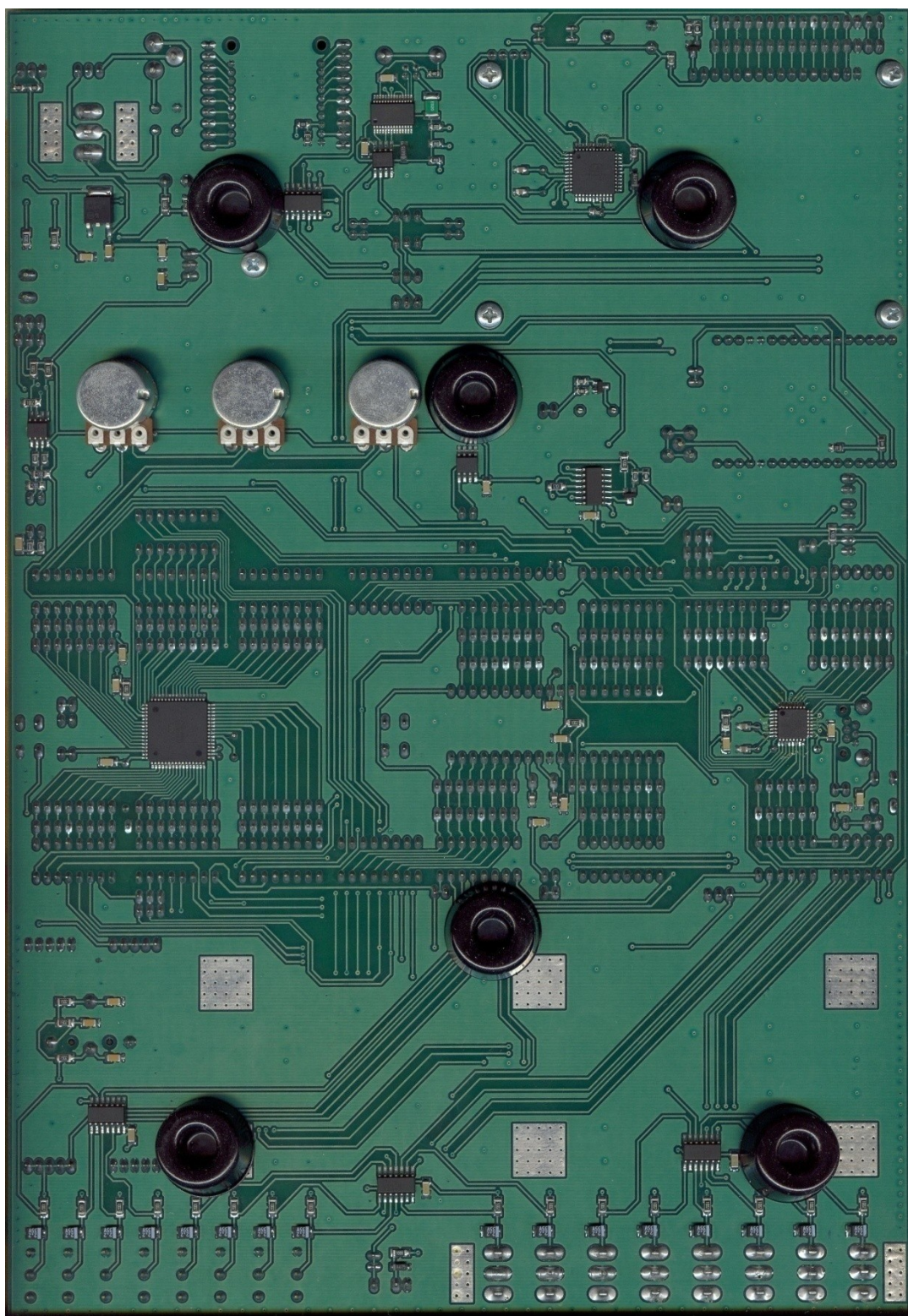


Obr. 60 - Potisk vývojové desky

VII. Foto vývojové desky

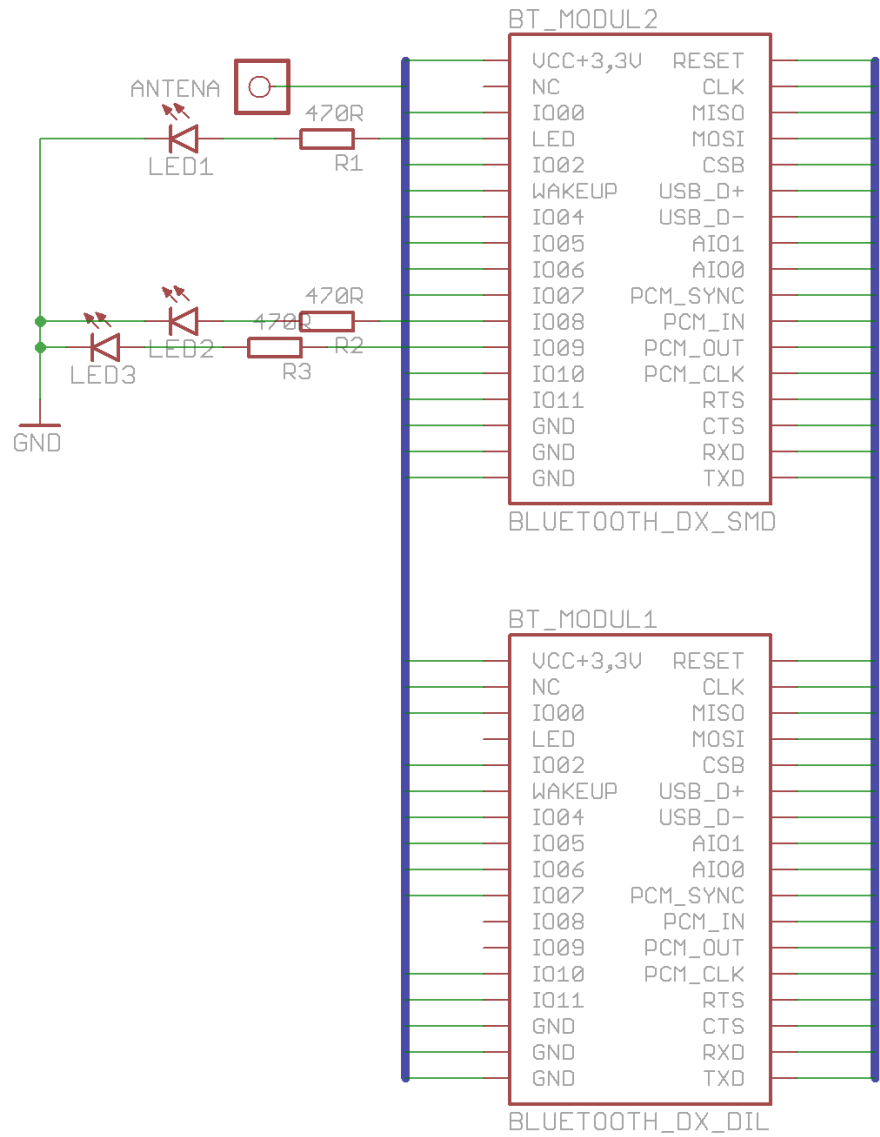


Obr. 61 - Vývojová deska strana TOP



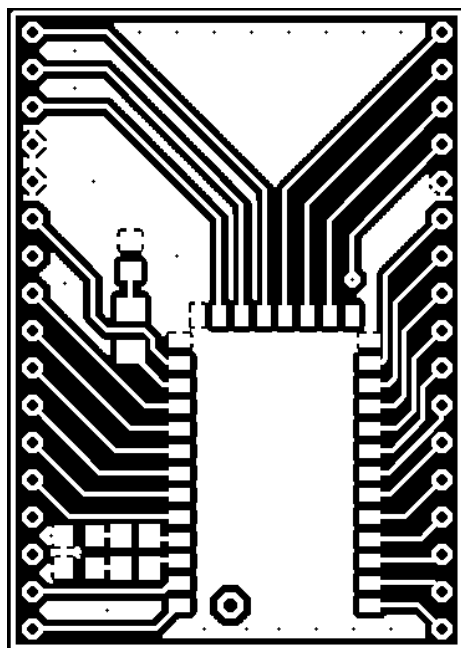
Obr. 62 - Vývojová deska strana BOTTOM

VIII. Elektrické schéma redukce bluetooth modulu

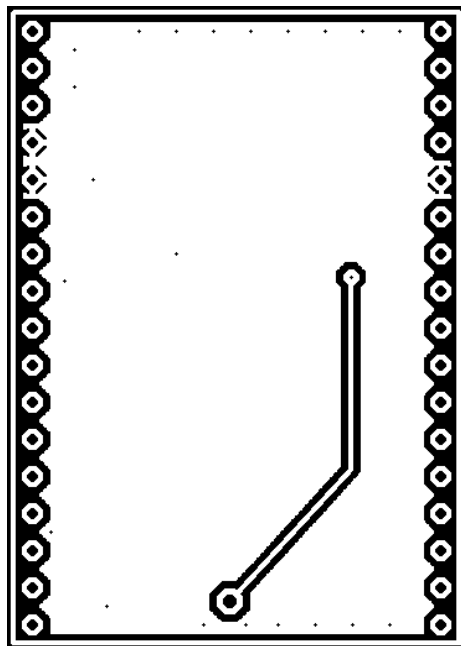


Obr. 63 - Schéma redukce bluetooth modulu

IX. DPS redukce bluetooth modulu

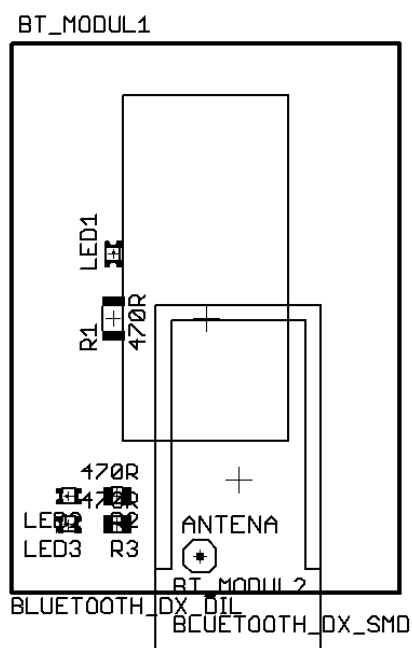


Obr. 64 - DPS bluetooth modulu strana TOP



Obr. 65 - DPS bluetooth modulu strana BOTTOM

X. Osazovací plán redukce bluetooth modulu

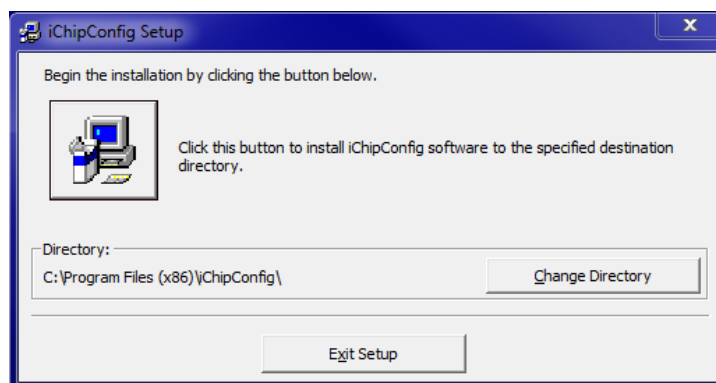


Obr. 66 - Osazovací plán bluetooth modulu

XI. Instalace nezbytného software

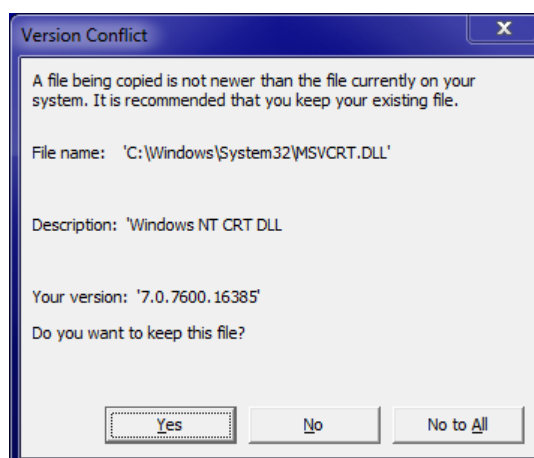
Instalace iChipConfig

Po spuštění instalace budeme upozorněni, že je nejprve nutné provést restart počítače. Po restartu nás povede průvodce instalací a vyzve nás k zadání adresáře, do kterého chceme umístit tento program.



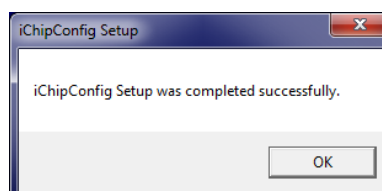
Obr. 67 - Instalace programu iChipConfig

Následně budeme několikrát upozorněni, že některé knihovny již v počítači jsou a že se je snažíme nahradit starším typem. Potvrdíme tedy tlačítkem Yes pro zachování novějších knihoven.



Obr. 68 - Instalace programu iChipConfig

Dokončení instalace reprezentuje hlášení o úspěšném dokončení.



Obr. 69 - Instalace programu iChipConfig

Instalace HW VSP3

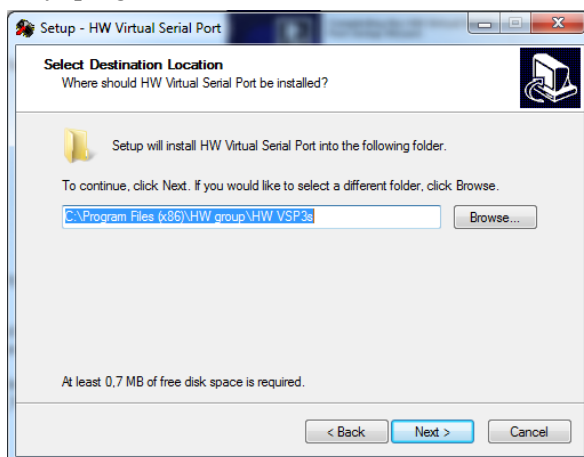
Krok 1: Spustit instalační program HW VSP3 single.

Krok 2: Zobrazí se uvítací okno, kde lze zjistit instalovanou verzi programu.

Krok 3: Odsouhlasením licenčních podmínek se povolí další instalace.

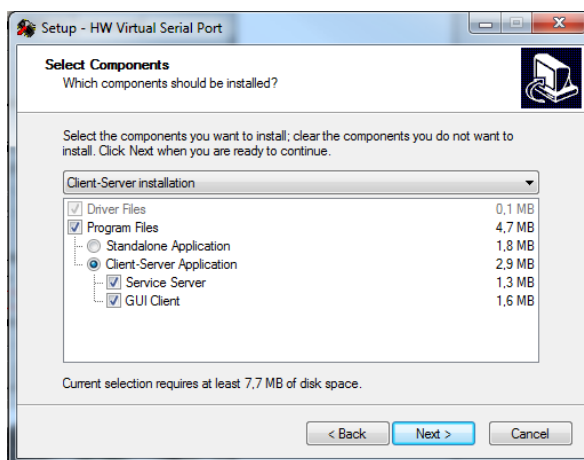
Krok 4: Zobrazí se základní informace o produktu.

Krok 5: Zvolte místo kam má být program nainstalován.



Obr. 70 - Instalace programu HW VSP3

Krok 6: Zvolte typ instalace Client – Server.



Obr. 71 - Instalace programu HW VSP3

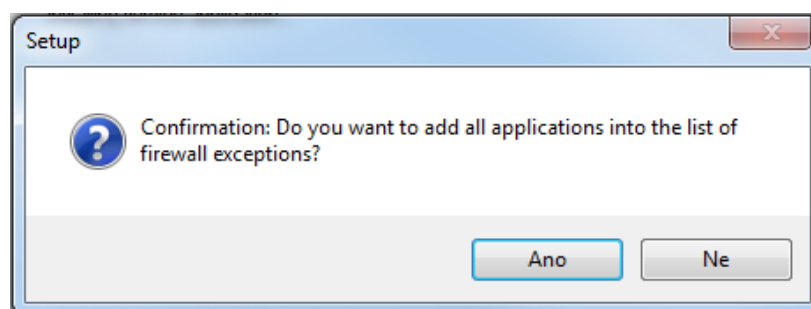
Krok 7: Zvolte název složky, která bude vytvořena v nabídce Start.

Krok 8: Zvolte, zda má být vytvořen zástupce programu na ploše.

Krok 9: Zobrazí se zvolené údaje, je doporučeno je zkontrolovat.

Krok 10: Potvrďte samotnou instalaci programu.

Krok 11: HW VSP vás vyzve k souhlasu se zařazením HW VSP do seznamu výjimek firewallu systému Windows. Souhlas je nezbytný pro správný běh aplikace. Nedáte-li ho, budete muset aplikaci i službu přidat do výjimek ručně.



Obr. 72 - Instalace programu HW VSP3

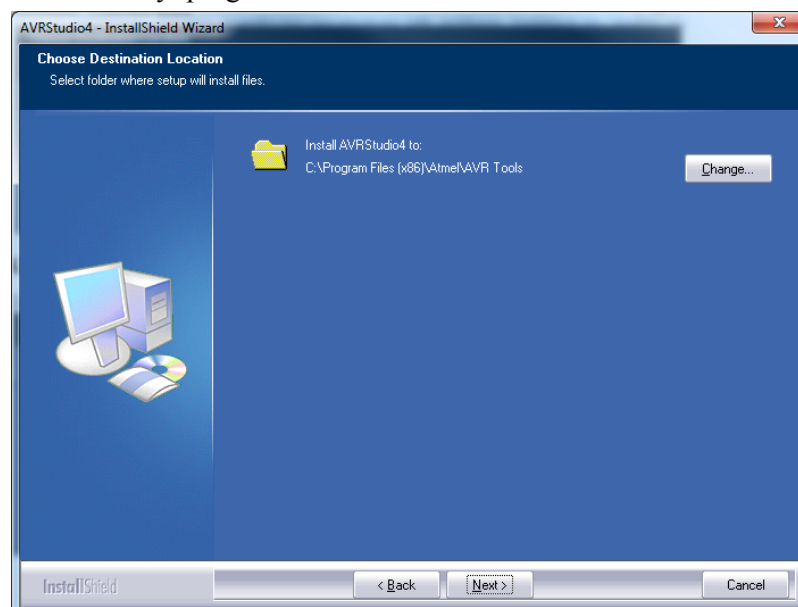
Krok 12: Instalaci dokončíte stiskem tlačítka Finish.[16]

Instalace AVR Studio

Krok 1: Spustit instalační program AVR studio 4.

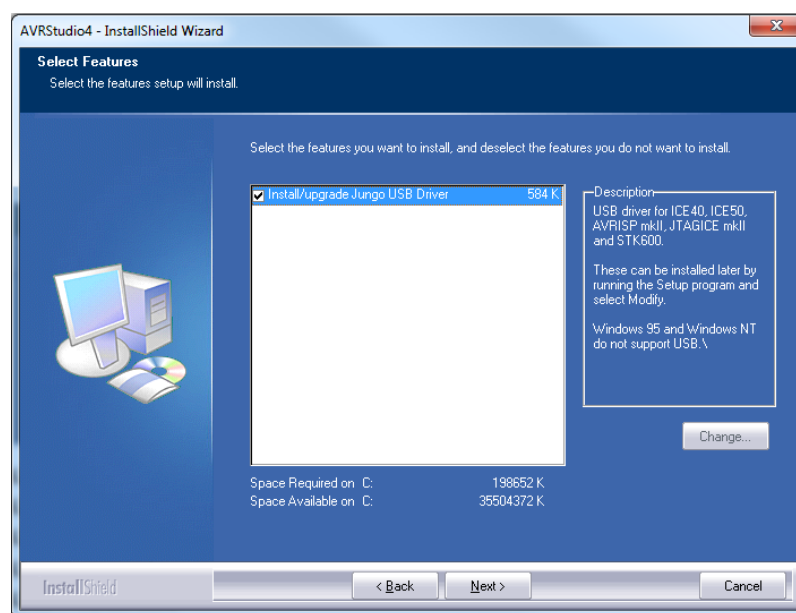
Krok 2: Odsouhlasit licenční podmínky.

Krok 3: Zvolte místo kam má být program nainstalován.



Obr. 73 - Instalace vývojového prostředí AVR Studio

Krok 4: Zvolit instalaci nebo aktualizaci Jungo USB ovladače.



Obr. 74 - Instalace vývojového prostředí AVR Studio

Krok 5: Kliknout na Install.

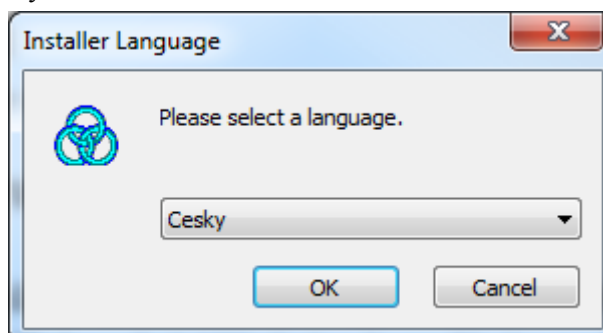
Krok 6: Instalaci dokončíte stiskem tlačítka Finish.

Instalace WinAVR

Jedná se o kompilátor jazyka C pro procesory z řady AVR. Jeho základ je postaven na kompilátoru GCC. WinAVR je volně šiřitelný a je vydáván pod licencí GNU, což znamená, že ho může kdokoliv šířit a upravovat, pokud uvede původní zdroj.

Krok 1: Spustit instalační program WinAVR.

Krok 2: Vybrat instalační jazyk.

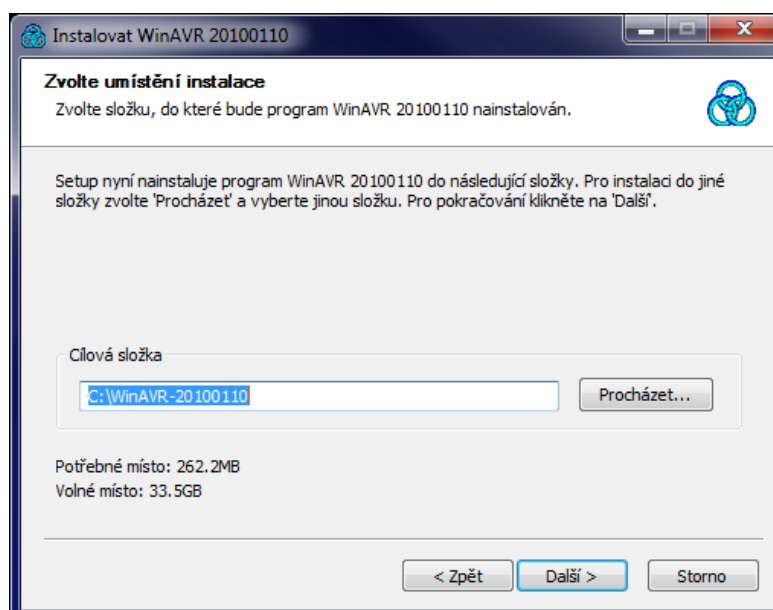


Obr. 75 - Instalace kompilátoru jazyka C

Krok 3: Zobrazí se uvítací okno, kde lze zjistit instalovanou verzi programu.

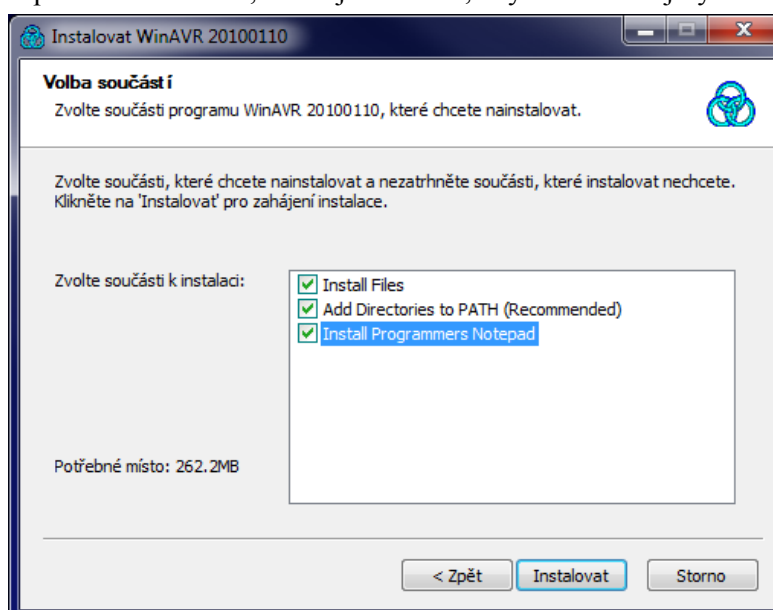
Krok 4: Odsouhlasit licenční podmínky.

Krok 5: Vybrat cestu kam se má nainstalovat WinAVR. Defaultně je vybrána cesta přímo do kořenového adresáře C:\ pokud tuto cestu změníme AVR Studio tento překladač nenalezne a nepůjde zkompileovat vytvořený projekt.



Obr. 76 - Instalace kompilátoru jazyka C

Krok 6: Instalace notepadu není důležitá, avšak je užitečná, když není v PC jiný editor zdrojového textu.



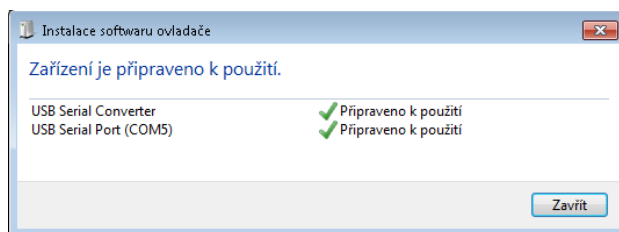
Obr. 77 - Instalace kompilátoru jazyka C

Krok 7: Kliknout na Instalovat

Krok 8: Instalaci dokončíme kliknutím na Dokončit

Instalace ovladačů FT232

Při použití operačního systému windows 7 a připojení vývojové desky do USB portu se automaticky začne prohledávat web windows update a potřebné ovladače se nainstalují samy. Pokud by však PC nebyl připojen k internetu lze nastavit cestu k ovladačům na přiloženém CD.



Obr. 78 - Instalace ovladačů FT232

XII. Knihovny pro program Eagle

Nejprve vytvoříme knihovny součástek. V menu File příkaz New a vybereme Library. Takto vytvořenou knihovnu uložíme příkazem File - Save. Zde knihovnu pojmenuje podle toho, jaké součástky bude obsahovat. Po uložení všech knihoven vytvoříme v každé knihovně námi požadované součástky.

Vytvoření nové součástky v knihovně má 3 kroky. Tyto kroky musíme zachovat, protože navazují jeden na druhý.

Začneme otevřením knihovny. Příkaz Open z menu File nebo příkaz New.

Vytvoření pouzdra:

Pouzdro (Package) je částí součástky (Device), která se přenáší na desku. Klikneme na ikonu Edit Package a editujeme nové pouzdro napsáním jeho jména do pole New v dialogovém rámečku. Nastavíme odpovídající rastr vývodů příkazem GRID.

Umístíme odpovídající pájecí plošky příkazem PAD a pojmenujeme je příkazem NAME.

Vložíme texty >NAME a >VALUE příkazem TEXT do odpovídajících kreslicích hladin (ukazují pak skutečné jméno a hodnotu součástky na desce v potisku) a nakreslíme obrys součástky příkazem WIRE v odpovídající kreslicí hladině (tPlace nebo bPlace - obrys z horní a dolní strany desky - lze pak použít při potisku desky).

Vytvoření schematické značky:

Schematická značka (dále nazývaná symbol) je část součástky, která se umísťuje do schématu. Klikneme na ikonu Edit Symbol a editujeme novou značku napsáním jejího jména do pole New v dialogovém rámečku.

Umístíme a pojmenujeme vývody značky (dále nazývané piny) příkazy PIN a NAME a přiřadíme pinům parametry příkazem (CHANGE).

Vložíme texty >NAME a >VALUE příkazem TEXT do odpovídajících kreslicích hladin (ukazují pak skutečné jméno a hodnotu součástky ve schématu) a nakreslíme obrysy symbolu příkazem WIRE v příslušné kreslicí hladině.

Vytvoření součástky:

Součástky (Device) jsou "řídícími" částmi celé součástky v knihovně a skládají se z pouzdra a jednoho nebo více symbolů.

Klikneme na ikonu Edit Device a editujeme novou součástku napsáním jejího jména v poli New dialogového rámečku.

Přiřadíme součástce pouzdro příkazem PACKAGE.

Přidáme do součástky symbol(y) (hradla) příkazem ADD, můžeme přidat tolik symbolů, kolik je za potřebí.

Příkazem CONNECT definujeme vzájemná propojení mezi piny symbolů a vývody (pady) pouzdra.

Uložíme knihovnu na disk a nyní můžeme používat novou součástku příkazem USE z editoru schémat nebo desky. [23]

Name	Description
Libraries	
#Mipi_knihovny	
#Mipi_chladice.lbr	Chladice
#Mipi_IO.lbr	IO, Akcelerometry, atd...
#Mipi_konektory.lbr	Konektory, piny a dalsi spojovací prvky.
#Mipi_moduly.lbr	Hotove moduly
#Mipi_montazni_diry.lbr	Montazni diry
#Mipi_napajeni.lbr	Napajeci symboly
#Mipi_ostatni.lbr	Ostatni soucastky
#Mipi_ramecky.lbr	Ramecky
#Mipi_RLC.lbr	Rezistory, Cívky a kondenzatory
#Mipi_spinaci_prvky.lbr	Tlacitka, Vypinace, atd...
#Mipi_stab_diody.lbr	Stabilizatory, diodove mustky, atd...
#Mipi_tranzistory.lbr	Tranzistory
#Mipi_u-Procesory.lbr	Mikrokontrolery
#Mipi_zobrazovace.lbr	LCD displeje, diody, 7-segmentovky, atd...

Obr. 79 - Knihovny v programu Eagle

XIII. Menu pro program Eagle

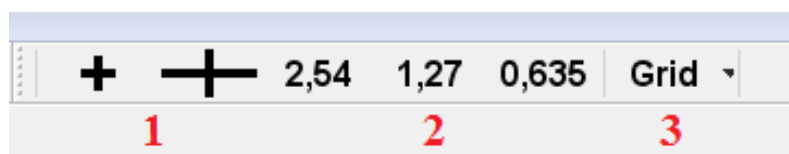
Návrhový program Eagle patří do skupiny jednoduchých a méně propracovaných softwareů. Velké množství používaných funkcí se nachází na nesmyslných místech nebo úplně chybí. Zobrazování jednotlivých stran TOP a BOTTOM se provádí pomocí vypínání jednotlivých vrstev a neexistuje možnost jednoduchého přepínání. Velká výhoda tohoto software je nízká cena. Při omezených rozměrech desky a vrstev je zdarma.

Tento problém se dá vyřešit napsáním scriptu a poté jeho jednoduchým spouštěním. Lze také vytvořit menu, pomocí kterého se tyto scripty spouští pouhým kliknutím na obrázek.

V adresáři C:\Program Files (x86)\EAGLE-5.11.0\scr se nachází soubor eagle.scr. Jedná se o soubor, který se automaticky spouští při otevření programu. Nachází se zde příkazy, které se provedou ihned po spuštění.

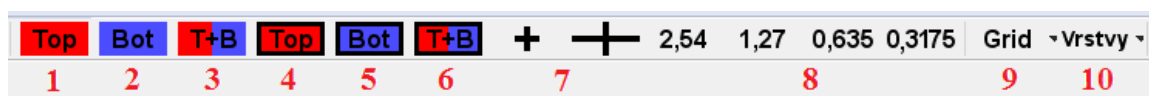
Pokud chceme automaticky při spuštění editoru schémat spustit nějaký příkaz, provedeme to pomocí příkazu SCH: a následným zapsáním příkazů oddělených středníkem. Každý editor má svůj specifický spouštěcí příkaz.

Ikonové menu se vytváří příkazem MENU a vypsáním příkazů v uvozovkách zakončený zpětným lomítkem. Každá položka může mít obrázkovou ikonu nebo jen nápis. Při použití obrázku je jeho velikost 22 px * 44 px a nachází se v adresáři C:\Program Files (x86)\EAGLE-5.11.0\bin.



Obr. 80 - Menu v editoru schémat

- 1 – malým křížkem se zvolí kurzor myši cca 0,5 cm a velkým se zobrazí kurzor přes celou obrazovku,
- 2 – hodnota udává velikost rastru v mm,
- 3 – méně používané rozteče rastru a možnost zapnutí/vypnutí se volí z roletového menu Grid



Obr. 81 - Menu v editoru DPS

- 1 – zobrazí se horní vrstva mědi, via a pady,
- 2 – zobrazí se spodní vrstva mědi, via a pady,
- 3 – zobrazí se spodní vrstva mědi, horní vrstva mědi, via a pady,
- 4 – zobrazí se horní vrstva mědi, horní meze součástek via a pady,
- 5 – zobrazí se spodní vrstva mědi, spodní meze součástek, via a pady,
- 6 – zobrazí se spodní a horní vrstva mědi, spodní a horní meze součástek, via a pady,
- 7 – malým křížkem se zvolí kurzor myši cca 0,5 cm a velkým se zobrazí kurzor přes celou obrazovku,
- 8 – hodnota udává velikost rastru v mm,
- 9 – méně používané rozteče rastru a možnost zapnutí/vypnutí se volí z roletového menu Grid,
- 10 – méně používané zapínání/vypínání vrstev, zobrazení osazovacího plánu desky a nespojené cesty se nacházejí v roletovém menu Vrstvy